ADVANCE INFORMATION

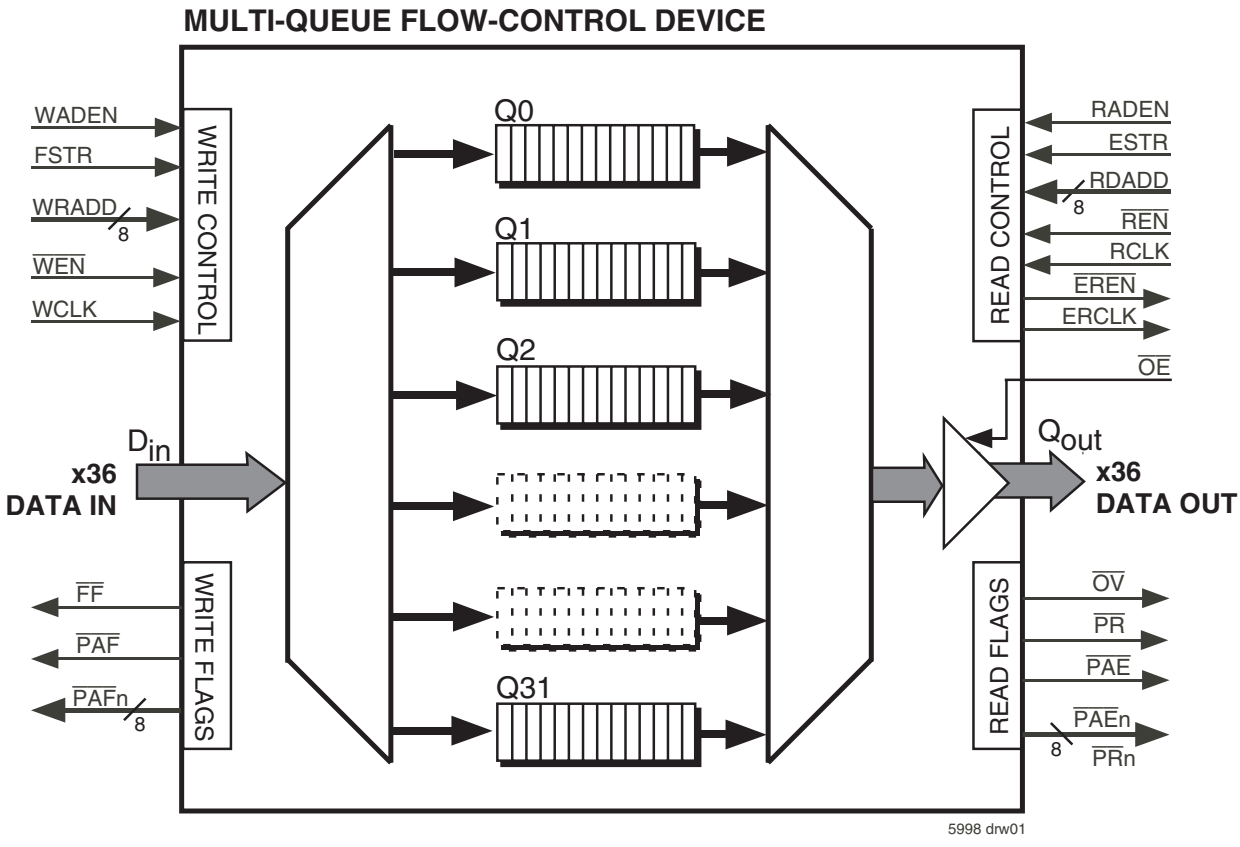
2.5V MULTI-QUEUE FLOW-CONTROL DEVICES (32 QUEUES) 36 BIT WIDE CONFIGURATION 1,179,648 bits and 2,359,296 bits

IDT72T51546
IDT72T51556

FEATURES:

- Choose from among the following memory density options:
  IDT72T51546 — Total Available Memory = 1,179,648 bits
  IDT72T51556 — Total Available Memory = 2,359,296 bits
- Configurable from 1 to 32 Queues
- Queues may be configured at master reset from the pool of Total Available Memory in blocks of 256 x 36
- Independent Read and Write access per queue
- User programmable via serial port
- User selectable I/O: 2.5V LVTTL, 1.5V HSTL, 1.8V eHSTL
- Default multi-queue device configurations
  – *IDT72T51546 : 1,024 x 36 x 32Q*
  – *IDT72T51556 : 2,048 x 36 x 32Q*
- 100% Bus Utilization, Read and Write on every clock cycle
- 200 MHz High speed operation (5ns cycle time)
- 3.6ns access time
- Echo Read Enable & Echo Read Clock Outputs
- Individual, Active queue flags ($\overline{OV}$, $\overline{FF}$, $\overline{PAE}$, $\overline{PAF}$, $\overline{PR}$)
- 8 bit parallel flag status on both read and write ports
- Shows $\overline{PAE}$ and $\overline{PAF}$ status of 8 Queues
- Direct or polled operation of flag status bus
- Global Bus Matching - (All Queues have same Input Bus Width and Output Bus Width)
- User Selectable Bus Matching Options:
  – *x36in to x36out*
  – *x18in to x36out*
  – *x9in to x36out*
  – *x36in to x18out*
  – *x36in to x9out*
- FWFT mode of operation on read port
- Packet mode operation
- Partial Reset, clears data in single Queue
- Expansion of up to 8 multi-queue devices in parallel is available
- Power Down Input provides additional power savings in HSTL and eHSTL modes.
- JTAG Functionality (Boundary Scan)
- Available in a 256-pin PBGA, 1mm pitch, 17mm x 17mm
- HIGH Performance submicron CMOS technology
- Industrial temperature range (-40°C to +85°C) is available

FUNCTIONAL BLOCK DIAGRAM



DESCRIPTION:

The IDT72T51546/72T51556 multi-queue flow-control devices is a single chip within which anywhere between 1 and 32 discrete FIFO queues can be setup. All queues within the device have a common data input bus, (write port) and a common data output bus, (read port). Data written into the write port is directed to a respective queue via an internal de-multiplex operation, addressed by the user. Data read from the read port is accessed from a respective queue via an internal multiplex operation, addressed by the user. Data writes and reads can be performed at high speeds up to 200MHz, with access times of 3.6ns. Data write and read operations are totally independent of each other, a queue maybe selected on the write port and a different queue on the read port or both ports may select the same queue simultaneously.

The device provides Full flag and Output Valid flag status for the queue selected for write and read operations respectively. Also a Programmable Almost Full and Programmable Almost Empty flag for each queue is provided. Two 8 bit programmable flag busses are available, providing status of queues not selected for write or read operations. When 8 or less queues are configured in the device these flag busses provide an individual flag per queue, when more than 8 queues are used, either a Polled or Direct mode of bus operation provides the flag busses with all queues status.

Bus Matching is available on this device, either port can be 9 bits, 18 bits or 36 bits wide provided that at least one port is 36 bits wide. When Bus Matching is used the device ensures the logical transfer of data throughput in a Little Endian manner.

A packet mode of operation is also provided when the device is configured for 36 bit input and 36 bit output port sizes. The Packet mode provides the user with a flag output indicating when at least one (or more) packets of data within a queue is available for reading. The Packet Ready provides the user with a means by which to mark the start and end of packets of data being passed through the queues. The multi-queue device then provides the user with an internally generated packet ready status per queue.

The user has full flexibility configuring queues within the device, being able to program the total number of queues between 1 and 32, the individual queue depths being independent of each other. The programmable flag positions are also user programmable. All programming is done via a dedicated serial port. If the user does not wish to program the multi-queue device, a default option is available that configures the device in a predetermined manner.

Both Master Reset and Partial Reset pins are provided on this device. A Master Reset latches in all configuration setup pins and must be performed before programming of the device can take place. A Partial Reset will reset the read and write pointers of an individual queue, provided that the queue is selected on both the write port and read port at the time of partial reset.

Echo Read Enable, $\overline{EREN}$ and Echo Read Clock, ERCLK outputs are provided. These are outputs from the read port of the Queue that are required for high speed data communication, to provide tighter synchronization between the data being transmitted from the Qn outputs and the data being received by the input device. Data read from the read port is available on the output bus with respect to $\overline{EREN}$ and ERCLK, this is very useful when data is being read at high speed.

The multi-queue flow-control device has the capability of operating its IO in either 2.5V LVTTL, 1.5V HSTL or 1.8V eHSTL mode. The type of IO is selected via the IOSEL input. The core supply voltage (V_{CC}) to the multi-queue is always 2.5V, however the output levels can be set independently via a separate supply, V_{DDQ}.

The devices also provide additional power savings via a Power Down Input. This input disables the write port data inputs when no write operations are required.

A JTAG test port is provided, here the multi-queue flow-control device has a fully functional Boundary Scan feature, compliant with IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture.

See Figure 1, *Multi-Queue Flow-Control Device Block Diagram* for an outline of the functional blocks within the device.

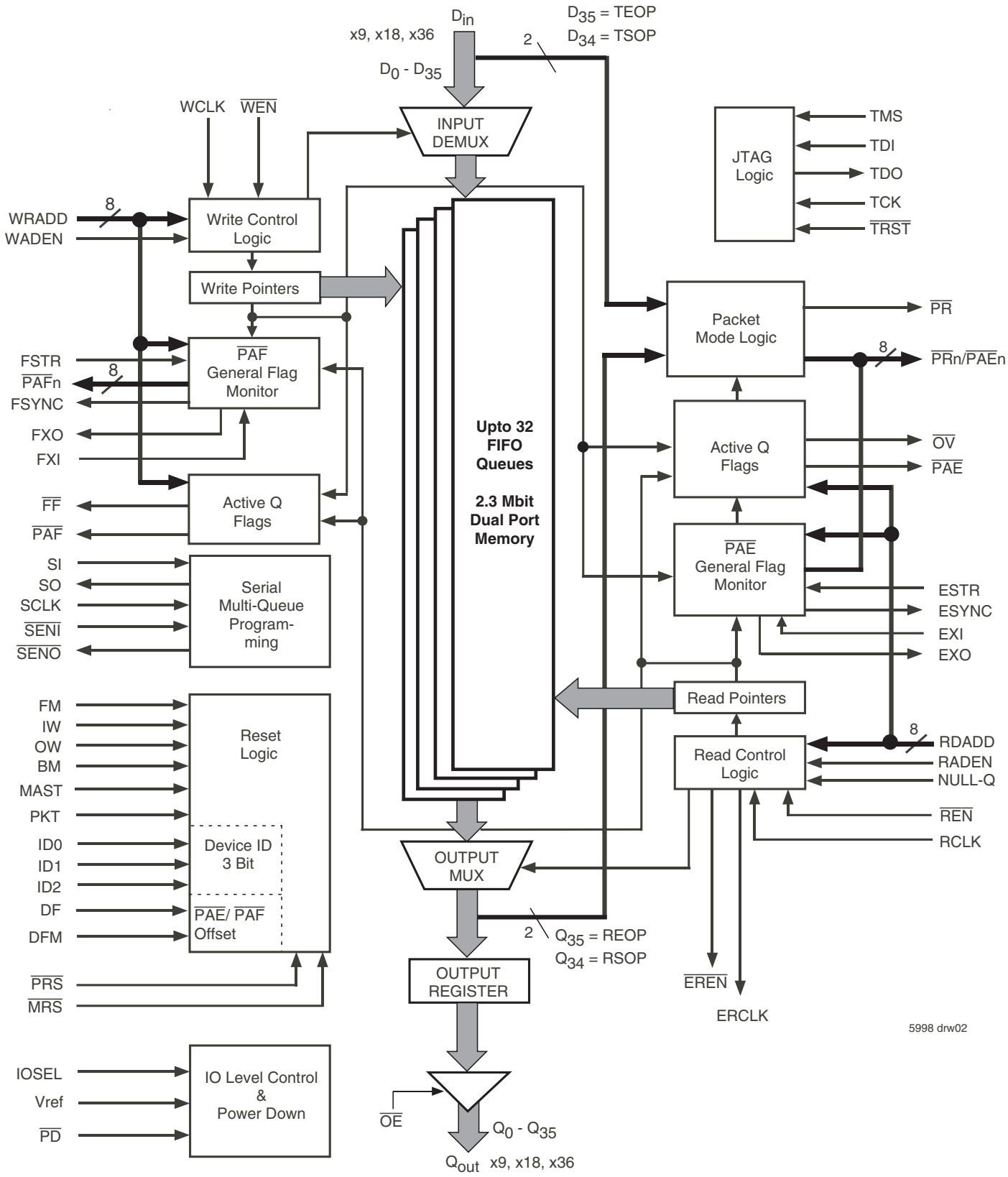
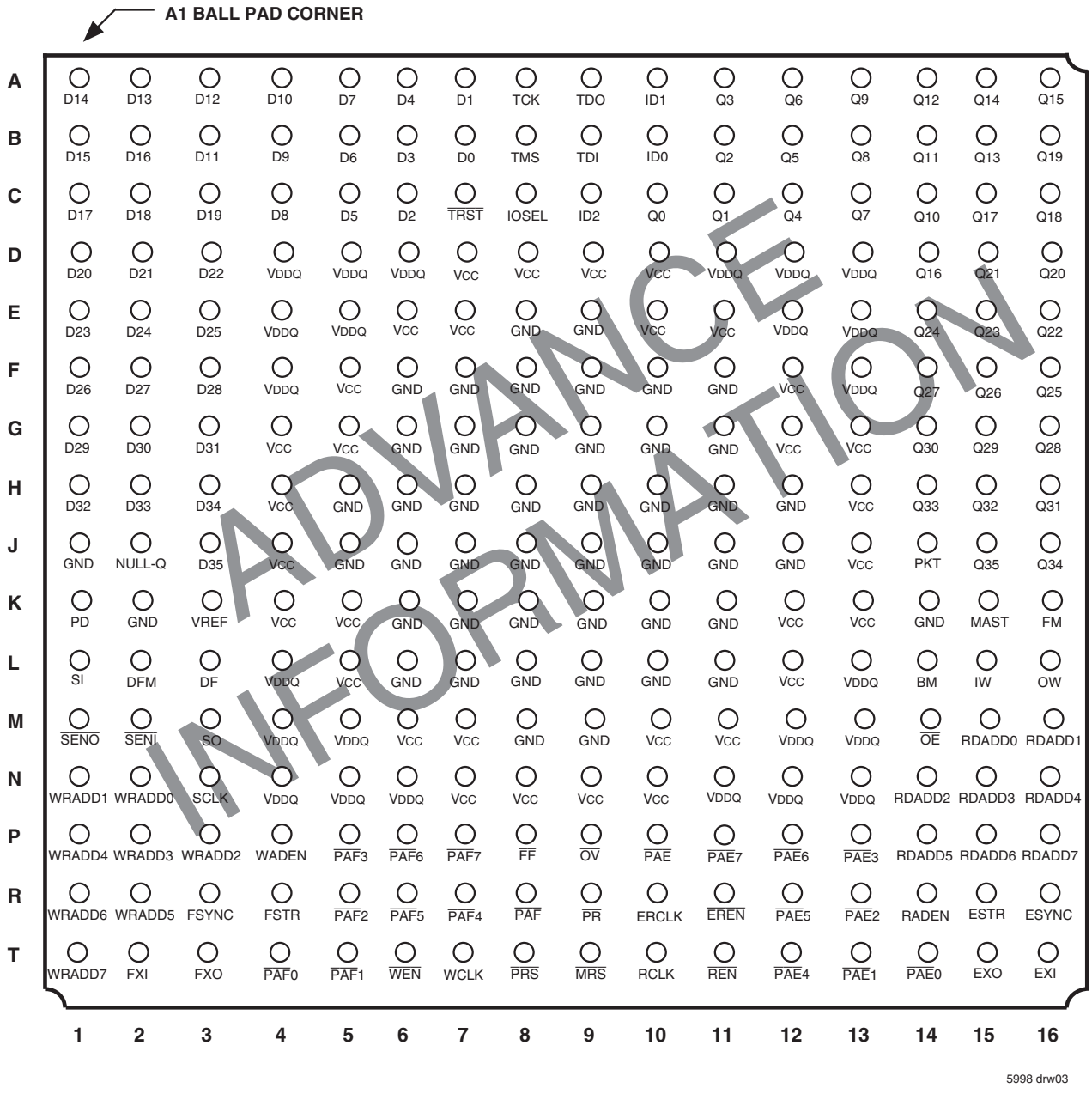


Figure 1. Multi-Queue Flow-Control Device Block Diagram

PIN CONFIGURATION



5998 drw03

PBGA (BB256-1, order code: BB)
TOP VIEW

DETAILED DESCRIPTION

MULTI-QUEUE STRUCTURE

The IDT multi-queue flow-control device has a single data input port and single data output port with up to 32 FIFO queues in parallel buffering between the two ports. The user can setup between 1 and 32 Queues within the device. These queues can be configured to utilize the total available memory, providing the user with full flexibility and ability to configure the queues to be various depths, independent of one another.

MEMORY ORGANIZATION/ ALLOCATION

The memory is organized into what is known as "blocks", each block being 256 x36 bits. When the user is configuring the number of queues and individual queue sizes the user must allocate the memory to respective queues, in units of blocks, that is, a single queue can be made up from 0 to m blocks, where m is the total number of blocks available within a device. Also the total size of any given queue must be in increments of 256 x36. For the IDT72T51546 and IDT72T51556 the Total Available Memory is 128 and 256 blocks respectively (a block being 256 x36). Queues can be built from these blocks to make any size queue desired and any number of queues desired.

BUS WIDTHS

The input port is common to all queues within the device, as is the output port. The device provides the user with Bus Matching options such that the input port and output port can be either x9, x18 or x36 bits wide provided that at least one of the ports is x36 bits wide, the read and write port widths being set independently of one another. Because the ports are common to all queues the width of the queues is not individually set, so that the input width of all queues are equal and the output width of all queues are equal.

WRITING TO & READING FROM THE MULTI-QUEUE

Data being written into the device via the input port is directed to a discrete queue via the write queue select address inputs. Conversely, data being read from the device read port is read from a queue selected via the read queue select address inputs. Data can be simultaneously written into and read from the same queue or different queues. Once a queue is selected for data writes or reads, the writing and reading operation is performed in the same manner as a conventional IDT synchronous FIFO, utilizing clocks and enables, there is a single clock and enable per port. When a specific queue is addressed on the write port, data placed on the data inputs is written to that queue sequentially based on the rising edge of a write clock provided setup and hold times are met. Conversely, data is read on to the output port after an access time from a rising edge on a read clock.

The operation of the write port is comparable to the function of a conventional FIFO operating in standard IDT mode. Write operations can be performed on the write port provided that the queue currently selected is not full, a full flag output provides status of the selected queue. The operation of the read port is comparable to the function of a conventional FIFO operating in FWFT mode. When a queue is selected on the output port, the next word in that queue will automatically fall through to the output register. All subsequent words from that queue require an enabled read cycle. Data cannot be read from a selected queue if that queue is empty, the read port provides an Output Valid flag indicating when data read out is valid. If the user switches to a queue that is empty, the last word from the previous queue will remain on the output register.

As mentioned, the write port has a full flag, providing full status of the selected queue. Along with the full flag a dedicated almost full flag is provided, this almost full flag is similar to the almost full flag of a conventional IDT FIFO. The device provides a user programmable almost full flag for all 32 queues and when a respective queue is selected on the write port, the almost full flag provides status for that queue. Conversely, the read port has an output valid flag, providing status of the data being read from the queue selected on the read port. As well as the output valid flag the device provides a dedicated almost empty flag. This almost empty flag is similar to the almost empty flag of a conventional IDT FIFO. The device provides a user programmable almost empty flag for all 32 queues and when a respective queue is selected on the read port, the almost empty flag provides status for that queue.

PROGRAMMABLE FLAG BUSSES

In addition to these dedicated flags, full & almost full on the write port and output valid & almost empty on the read port, there are two flag status busses. An almost full flag status bus is provided, this bus is 8 bits wide. Also, an almost empty flag status bus is provided, again this bus is 8 bits wide. The purpose of these flag busses is to provide the user with a means by which to monitor the data levels within queues that may not be selected on the write or read port. As mentioned, the device provides almost full and almost empty registers (programmable by the user) for each of the 32 queues in the device.

In the IDT72T51546/72T51556 multi-queue flow-control devices the user has the option of utilizing anywhere between 1 and 32 queues, therefore the 8 bit flag status busses are multiplexed between the 32 queues, a flag bus can only provide status for 8 of the 32 queues at any moment, this is referred to as a "Quadrant", such that when the bus is providing status of queues 1 through 8, this is quadrant 1, when it is queues 9 through 16, this is quadrant 2 and so on up to quadrant 4. If less than 32 queues are setup in the device, there are still 4 quadrants, such that in "Polled" mode of operation the flag bus will still cycle through 4 quadrants. If for example only 22 queues are setup, quadrants 1 and 2 will reflect status of queues 1 through 8 and 9 through 16 respectively. Quadrant 3 will reflect the status of queues 17 through 22 on the least significant 6 bits, the most significant 2 bits of the flag bus are don't care and the 4th quadrant outputs will be don't care also.

The flag busses are available in two user selectable modes of operation, "Polled" or "Direct". When operating in polled mode a flag bus provides status of each quadrant sequentially, that is, on each rising edge of a clock the flag bus is updated to show the status of each quadrant in order. The rising edge of the write clock will update the almost full bus and a rising edge on the read clock will update the almost empty bus. The mode of operation is always the same for both the almost full and almost empty flag busses. When operating in direct mode, the quadrant on the flag bus is selected by the user. So the user can actually address the quadrant to be placed on the flag status busses, these flag busses operate independently of one another. Addressing of the almost full flag bus is done via the write port and addressing of the almost empty flag bus is done via the read port.

PACKET READY

The multi-queue flow-control device also offers a "Packet Mode" operation. Packet Mode is user selectable and requires the device to be configured with both write and read ports as 36 bits wide. In packet mode, users can define the length of packets or frame by using the two most significant bits of the 36-bit word. Bit 34 is used to mark the Start of Packet (SOP) and bit 35 is used to mark the End of Packet (EOP) as shown in Table 5). When writing data into a given queue , the first word being written is marked, by the user setting bit 34 as the "Start of Packet" (SOP) and the last word written is marked as the "End of Packet" (EOP) with all words written between the Start of Packet (SOP) marker (bit 34) and the End of packet (EOP) packet marker (bit 35) constituting the entire packet. A packet can be any length the user desires, up to the total available memory in the multi-queue device. The device monitors the SOP (bit 34) and looks for the word that contains the EOP (bit 35). The read port is supplied with an additional

status flag, "Packet Ready". The Packet Ready ($\overline{PR}$) flag in conjunction with Output Valid ($\overline{OV}$) indicates when at least one packet is available to read. When in packet mode the almost empty flag status , provides packet ready flag status for individual queues.

EXPANSION

Expansion of multi-queue devices is also possible, up to 8 devices can be connected in a parallel fashion providing the possibility of both depth expansion or queue expansion. Depth Expansion means expanding the depths of individual queues. Queue expansion means increasing the total number of queues available. Depth expansion is possible by virtue of the fact that more memory blocks within a multi-queue device can be allocated to increase the depth of a queue. For example, depth expansion of 8 devices provides the possibility of 8 queues of 64K x36 deep, each queue being setup within a single device utilizing all memory blocks available to produce a single queue. This is the deepest queue that can setup within a device.

For queue expansion a maximum number of 256 (8 x 32) queues may be setup, each queue being 2K x36 deep, if less queues are setup, then more memory blocks will be available to increase queue depths if desired. When connecting multi-queue devices in expansion mode all respective input pins (data & control) and output pins (data & flags), should be "connected" together between individual devices.

PIN DESCRIPTIONS

Symbol & Pin No.	Name	I/O TYPE	Description
BM (L14)	Bus Matching	LVTTL INPUT	This pin is setup before Master Reset and must not toggle during any device operation. This pin is used along with IW and OW to setup the multi-queue flow-control device bus width. Please refer to Table 3 for details.
D[35:0] Din (See Pin No. table for details)	Data Input Bus	HSTL-LVTTL INPUT	These are the 36 data input pins. Data is written into the device via these input pins on the rising edge of WCLK provided that $\overline{WEN}$ is LOW. Note, that in Packet mode D32-D35 may be used as packet markers, please see packet ready functional discussion for more detail. Due to bus matching not all inputs may be used, any unused inputs should be tied LOW.
DF[1] (L3)	Default Flag	LVTTL INPUT	If the user requires default programming of the multi-queue device, this pin must be setup before Master Reset and must not toggle during any device operation. The state of this input at master reset determines the value of the $\overline{PAE}$/$\overline{PAF}$ flag offsets. If DF is LOW the value is 8, if DF is HIGH the value is 128.
DFM[1] (L2)	Default Mode	LVTTL INPUT	The multi-queue device requires programming after master reset. The user can do this serially via the serial port, or the user can use the default method. If DFM is LOW at master reset then serial mode will be selected, if HIGH then default mode is selected.
ERCLK (R10)	RCLK Echo	HSTL-LVTTL OUTPUT	Read Clock Echo output, this output generates a clock based on the read clock input, this is used for Source Synchronous clocking where the receiving devices utilizes the ERCLK to clock data output from the Queue.
$\overline{EREN}$ (R11)	$\overline{REN}$ Echo	HSTL-LVTTL OUTPUT	Read Enable Echo output, can be used in conjunction with the ERCLK output to load data output from the Queue into the receiving device.
ESTR (R15)	$\overline{PAE}$n Flag Bus Strobe	LVTTL INPUT	If direct operation of the $\overline{PAE}$n bus has been selected, the ESTR input is used in conjunction with RCLK and the RDADD bus to select a quadrant of queues to be placed on to the $\overline{PAE}$n bus outputs. A quadrant addressed via the RDADD bus is selected on the rising edge of RCLK provided that ESTR is HIGH. If Polled operations has been selected, ESTR should be tied inactive, LOW. Note, that a $\overline{PAE}$n flag bus selection cannot be made, (ESTR must NOT go active) until programming of the part has been completed and $\overline{SENO}$ has gone LOW.
ESYNC (R16)	$\overline{PAE}$n Bus Sync	HSTL-LVTTL OUTPUT	ESYNC is an output from the multi-queue device that provides a synchronizing pulse for the $\overline{PAE}$n bus during Polled operation of the $\overline{PAE}$n bus. During Polled operation each quadrant of queue status flags is loaded on to the $\overline{PAE}$n bus outputs sequentially based on RCLK. The first RCLK rising edge loads quadrant 1 on to $\overline{PAE}$n, the second RCLK rising edge loads quadrant 2 and so on. The fifth RCLK rising edge will again load quadrant 1. During the RCLK cycle that quadrant 1 of a selected device is placed on to the $\overline{PAE}$n bus, the ESYNC output will be HIGH. For all other quadrants of that device, the ESYNC output will be LOW.
EXI (T16)	$\overline{PAE}$n Bus Expansion In	LVTTL INPUT	The EXI input is used when multi-queue devices are connected in expansion mode and Polled $\overline{PAE}$n bus operation has been selected . EXI of device 'N' connects directly to EXO of device 'N-1'. The EXI receives a token from the previous device in a chain. In single device mode the EXI input must be tied LOW if the $\overline{PAE}$n bus is operated in direct mode. If the $\overline{PAE}$n bus is operated in polled mode the EXI input must be connected to the EXO output of the same device. In expansion mode the EXI of the first device should be tied LOW, when direct mode is selected.
EXO (T15)	$\overline{PAE}$n Bus Expansion Out	LVTTL OUTPUT	EXO is an output that is used when multi-queue devices are connected in expansion mode and Polled $\overline{PAE}$n bus operation has been selected . EXO of device 'N' connects directly to EXI of device 'N+1'. This pin pulses when device N has placed its final (4th) quadrant on to the $\overline{PAE}$n bus with respect to RCLK. This pulse (token) is then passed on to the next device in the chain 'N+1' and on the next RCLK rising edge the first quadrant of device N+1 will be loaded on to the $\overline{PAE}$n bus. This continues through the chain and EXO of the last device is then looped back to EXI of the first device. The ESYNC output of each device in the chain provides synchronization to the user of this looping event.
$\overline{FF}$ (P8)	Full Flag	HSTL-LVTTL OUTPUT	This pin provides the full flag output for the active Queue, that is, the queue selected on the input port for write operations, (selected via WCLK, WRADD bus and WADEN). On the WCLK cycle after a queue selection, this flag will show the status of the newly selected queue. Data can be written to this queue on the next cycle provided $\overline{FF}$ is HIGH. This flag has High-Impedance capability, this is important during expansion of devices, when the $\overline{FF}$ flag output of up to 8 devices may be connected together on a common line. The device with a queue selected takes control of the $\overline{FF}$ bus, all other devices place their $\overline{FF}$ output into High-Impedance. When a queue selection is made on the write port this output will switch from High-Impedance control on the next WCLK cycle. This flag is synchronized to WCLK.

PIN DESCRIPTIONS (CONTINUED)

Symbol & Pin No.	Name	I/O TYPE	Description
FM[1] (K16)	Flag Mode	HSTL-LVTTL INPUT	This pin is setup before a master reset and must not toggle during any device operation. The state of the FM pin during Master Reset will determine whether the $\overline{PAF}$n and $\overline{PAE}$n flag busses operate in either Polled or Direct mode. If this pin is HIGH the mode is Polled, if LOW then it will be Direct.
FSTR (R4)	$\overline{PAF}$n Flag Bus Strobe	LVTTL INPUT	If direct operation of the $\overline{PAF}$n bus has been selected, the FSTR input is used in conjunction with WCLK and the WRADD bus to select a quadrant of queues to be placed on to the $\overline{PAF}$n bus outputs. A quadrant addressed via the WRADD bus is selected on the rising edge of WCLK provided that FSTR is HIGH. If Polled operations has been selected, FSTR should be tied inactive, LOW. Note, that a $\overline{PAF}$n flag bus selection cannot be made, (FSTR must NOT go active) until programming of the part has been completed and $\overline{SENO}$ has gone LOW.
FSYNC (R3)	$\overline{PAF}$n Bus Sync	LVTTL OUTPUT	FSYNC is an output from the multi-queue device that provides a synchronizing pulse for the $\overline{PAF}$n bus during Polled operation of the $\overline{PAF}$n bus. During Polled operation each quadrant of queue status flags is loaded on to the $\overline{PAF}$n bus outputs sequentially based on WCLK. The first WCLK rising edge loads quadrant 1 on to $\overline{PAF}$n, the second WCLK rising edge loads quadrant 2 and so on. The fifth WCLK rising edge will again load quadrant 1. During the WCLK cycle that quadrant 1 of a selected device is placed on to the $\overline{PAF}$n bus, the FSYNC output will be HIGH. For all other quadrants of that device, the FSYNC output will be LOW.
FXI (T2)	$\overline{PAF}$n Bus Expansion In	LVTTL INPUT	The FXI input is used when multi-queue devices are connected in expansion mode and Polled $\overline{PAF}$n bus operation has been selected . FXI of device 'N' connects directly to FXO of device 'N-1'. The FXI receives a token from the previous device in a chain. In single device mode the FXI input must be tied LOW if the $\overline{PAF}$n bus is operated in direct mode. If the $\overline{PAF}$n bus is operated in polled mode the FXI input must be connected to the FXO output of the same device. In expansion mode the FXI of the first device should be tied LOW, when direct mode is selected.
FXO (T3)	$\overline{PAF}$n Bus Expansion Out	LVTTL OUTPUT	FXO is an output that is used when multi-queue devices are connected in expansion mode and Polled $\overline{PAF}$n bus operation has been selected . FXO of device 'N' connects directly to FXI of device 'N+1'. This pin pulses when device N has placed its final (4th) quadrant on to the $\overline{PAF}$n bus with respect to WCLK. This pulse (token) is then passed on to the next device in the chain 'N+1' and on the next WCLK rising edge the first quadrant of device N+1 will be loaded on to the $\overline{PAF}$n bus. This continues through the chain and FXO of the last device is then looped back to FXI of the first device. The FSYNC output of each device in the chain provides synchronization to the user of this looping event.
ID[2:0][1] ID2-C9 ID1-A10 ID0-B10	Device ID Pins	HSTL-LVTTL INPUT	For the 32Q multi-queue device the WRADD and RDADD address busses are 8 bits wide. When a queue selection takes place the 3 MSb's of this 8 bit address bus are used to address the specific device (the 5 LSb's are used to address the queue within that device). During write/read operations the 3 MSb's of the address are compared to the device ID pins. The first device in a chain of multi-queue's (connected in expansion mode), may be setup as '000', the second as '001' and so on through to device 8 which is '111', however the ID does not have to match the device order. In single device mode these pins should be setup as '000' and the 3 MSb's of the WRADD and RDADD address busses should be tied LOW. The ID[2:0] inputs setup a respective devices ID during master reset. These ID pins must not toggle during any device operation. Note, the device selected as the 'Master' does not have to have the ID of '000'.
IOSEL (C8)	IO Select	LVTTL INPUT	This pin is used to select either HSTL or 2.5V LVTTL operation for the I/O. If HSTL or eHSTL I/O are required then IOSEL should be tied HIGH. If LVTTL I/O are required then it should be tied LOW.
IW[1] (L15)	Input Width	LVTTL INPUT	This pin is used in conjunction with OW and BM to setup the input and output bus widths to be a combination of x9, x18 or x36, (providing that one port is x36).
MAST[1] (K15)	Master Device	HSTL-LVTTL INPUT	The state of this input at Master Reset determines whether a given device (within a chain of devices), is the Master device or a Slave. If this pin is HIGH, the device is the master if it is LOW then it is a Slave. The master device is the first to take control of all outputs after a master reset, all slave devices go to High-Impedance, preventing bus contention. If a multi-queue device is being used in single device mode, this pin must be set HIGH.
$\overline{MRS}$ (T9)	Master Reset	HSTL-LVTTL INPUT	A master reset is performed by taking $\overline{MRS}$ from HIGH to LOW, to HIGH. Device programming is required after master reset.
NULL-Q (J2)	Null Queue Select	HSTL-LVTTL INPUT	This pin is used on the read port when a Null-Q is required, it is used in conjunction with the RDADD and RADEN address bus to address the Null-Q.

PIN DESCRIPTIONS (CONTINUED)

Symbol & Pin No.	Name	I/O TYPE	Description
$\overline{OE}$ (M14)	Output Enable	HSTL-LVTTL INPUT	The Output enable signal is an Asynchronous signal used to provide three-state control of the multi-queue data output bus, Qout. If a device has been configured as a "Master" device, the Qout data outputs will be in a Low Impedance condition if the $\overline{OE}$ input is LOW. If $\overline{OE}$ is HIGH then the Qout data outputs will be in High Impedance. If a device is configured a "Slave" device, then the Qout data outputs will always be in High Impedance until that device has been selected on the Read Port, at which point $\overline{OE}$ provides three-state of that respective device.
$\overline{OV}$ (P9)	Output Valid Flag	HSTL-LVTTL OUTPUT	This output flag provides output valid status for the data word present on the multi-queue flow-control device data output port, Qout. This flag is therefore, 2-stage delayed to match the data output path delay. That is, there is a 2 RCLK cycle delay from the time a given queue is selected for reads, to the time the $\overline{OV}$ flag represents the data in that respective queue. When a selected queue on the read port is read to empty, the $\overline{OV}$ flag will go HIGH, indicating that data on the output bus is not valid. The $\overline{OV}$ flag also has High-Impedance capability, required when multiple devices are used and the $\overline{OV}$ flags are tied together.
OW[1] (L16)	Output Width	LVTTL INPUT	This pin is setup during Master Reset and must not toggle during any device operation. This pin is used in conjunction with IW and BM to setup the data input and output bus widths to be a combination of x9, x18 or x36, (providing that one port is x36).
$\overline{PAE}$ (P10)	Programmable Almost-Empty Flag	HSTL-LVTTL OUTPUT	This pin provides the Almost-Empty flag status for the Queue that has been selected on the output port for read operations, (selected via RCLK, RDADD and RADEN). This pin is LOW when the selected Queue is almost-empty. This flag output may be duplicated on one of the $\overline{PAE}$n bus lines. This flag is synchronized to RCLK.
$\overline{PAE}$n/$\overline{PR}$n ($\overline{PAE}$7-P11 $\overline{PAE}$6-P12 $\overline{PAE}$5-R12 $\overline{PAE}$4-T12 $\overline{PAE}$3-P13 $\overline{PAE}$2-R13 $\overline{PAE}$1-T13 $\overline{PAE}$0-T14)	Programmable Almost-Empty Flag Bus/Packet ReadyFlag Bus	HSTL-LVTTL OUTPUT	On the 32Q device the $\overline{PAE}$n/$\overline{PR}$n bus is 8 bits wide. During a Master Reset this bus is setup for either Almost Empty mode or Packet mode. This output bus provides $\overline{PAE}$/$\overline{PR}$n status of 8 queues (1 quadrant), within a selected device, having a total of 4 quadrants. During Queue read/write operations these outputs provide programmable empty flag status or packet ready status, in either direct or polled mode. The mode of flag operation is determined during master reset via the state of the FM input. This flag bus is capable of High-Impedance state, this is important during expansion of multi-queue devices. During direct operation the $\overline{PAE}$n/$\overline{PR}$n bus is updated to show the $\overline{PAE}$/$\overline{PR}$ status of a quadrant of queues within a selected device. Selection is made using RCLK, ESTR and RDADD. During Polled operation the $\overline{PAE}$n/$\overline{PR}$n bus is loaded with the $\overline{PAE}$/$\overline{PR}$n status of multi-queue flow-control quadrants sequentially based on the rising edge of RCLK. $\overline{PAE}$ or $\overline{PR}$ operation is determined by the state of PKT during master reset.
$\overline{PAF}$ (R8)	Programmable Almost-Full Flag	HSTL-LVTTL OUTPUT	This pin provides the Almost-Full flag status for the Queue that has been selected on the input port for write operations, (selected via WCLK, WRADD and WADEN). This pin is LOW when the selected Queue is almost-full. This flag output may be duplicated on one of the $\overline{PAF}$n bus lines. This flag is synchronized to WCLK.
$\overline{PAF}$n ($\overline{PAF}$7-P7 $\overline{PAF}$6-P6 $\overline{PAF}$5-R6 $\overline{PAF}$4-R7 $\overline{PAF}$3-P5 $\overline{PAF}$2-R5 $\overline{PAF}$1-T5 $\overline{PAF}$0-T4)	Programmable Almost-Full Flag Bus	HSTL-LVTTL OUTPUT	On the 32Q device the $\overline{PAF}$n bus is 8 bits wide. At any one time this output bus provides $\overline{PAF}$ status of 8 queues (1 quadrant), within a selected device, having a total of 4 quadrants. During Queue read/write operations these outputs provide programmable full flag status, in either direct or polled mode. The mode of flag operation is determined during master reset via the state of the FM input. This flag bus is capable of High-Impedance state, this is important during expansion of multi-queue devices. During direct operation the $\overline{PAF}$n bus is updated to show the $\overline{PAF}$ status of a quadrant of queues within a selected device. Selection is made using WCLK, FSTR, WRADD and WADEN. During Polled operation the $\overline{PAF}$n bus is loaded with the $\overline{PAF}$ status of multi-queue flow-control quadrants sequentially based on the rising edge of WCLK.
PD (K1)	Power Down	HSTL INPUT	This input is used to provide additional power savings. When the device I/O is setup for HSTL/eHSTL mode a HIGH on the PD input disables the data inputs on the write port only, providing significant power savings. In LVTTL mode this pin has no operation
PKT[1] (J14)	Packet Mode	LVTTL INPUT	The state of this pin during a Master Reset will determine whether the part is operating in Packet mode providing both a Packet Ready ($\overline{PR}$) output and a Programmable Almost Empty ($\overline{PAE}$) discrete output, or standard mode, providing a ($\overline{PAE}$) output only. If this pin is HIGH during Master Reset the part will operate in packet mode, if it is LOW then almost empty mode. If packet mode has been selected the read port flag bus becomes packet ready flag bus, $\overline{PR}$n and the discrete packet ready flag, $\overline{PR}$ is functional. If almost empty operation has been selected then the flag bus provides almost empty status, $\overline{PAE}$n and

PIN DESCRIPTIONS (CONTINUED)

Symbol & Pin No.	Name	I/O TYPE	Description
PKT[1] (Continued) (J14)	Packet Mode	LVTTL INPUT	the discrete almost empty flag, $\overline{PAE}$ is functional, the $\overline{PR}$ flag is inactive and should not be connected. Packet Ready utilizes user marked locations to identify start and end of packets being written into the device. Packet Mode can only be selected if both the input port width and output port width are 36 bits.
$\overline{PR}$ (R9)	Packet Ready Flag	HSTL-LVTTL OUTPUT	If packet mode has been selected this flag output provides Packet Ready status of the Queue selected for read operations. During a master reset the state of the PKT input determines whether Packet mode of operation will be used. If Packet mode is selected, then the condition of the $\overline{PR}$ flag and $\overline{OV}$ signal are asserted indicates a packet is ready for reading. The user must mark the start of a packet and the end of a packet when writing data into a queue. Using these Start Of Packet (SOP) and End Of Packet (EOP) markers, the multi-queue device sets $\overline{PR}$ LOW if one or more "complete" packets are available in the queue. A complete packet(s) must be written before the user is allowed to switch queues.
$\overline{PRS}$ (T8)	Partial Reset	HSTL-LVTTL INPUT	A Partial Reset can be performed on a single queue selected within the multi-queue device. Before a Partial Reset can be performed on a queue, that queue must be selected on both the write port and read port 2 clock cycles before the reset is performed. A Partial Reset is then performed by taking $\overline{PRS}$ LOW for one WCLK cycle and one RCLK cycle. The Partial Reset will only reset the read and write pointers to the first memory location, none of the devices configuration will be changed.
Q[35:0] Qout (See Pin No. table for details)	Data Output Bus	HSTL-LVTTL OUTPUT	These are the 36 data output pins. Data is read out of the device via these output pins on the rising edge of RCLK provided that $\overline{REN}$ is LOW, $\overline{OE}$ is LOW and the Queue is selected. Note, that in Packet Ready mode Q32-Q35 may be used as packet markers, please see packet ready functional discussion for more detail. Due to bus matching not all outputs may be used, any unused outputs should not be connected.
RADEN (R14)	Read Address Enable	HSTL-LVTTL INPUT	The RADEN input is used in conjunction with RCLK and the RDADD address bus to select a queue to be read from. A queue addressed via the RDADD bus is selected on the rising edge of RCLK provided that RADEN is HIGH. RADEN should be asserted (HIGH) only during a queue change cycle(s). RADEN should not be permanently tied HIGH. RADEN cannot be HIGH for the same RCLK cycle as ESTR. Note, that a read queue selection cannot be made, (RADEN must NOT go active) until programming of the part has been completed and $\overline{SENO}$ has gone LOW.
RCLK (T10)	Read Clock	HSTL-LVTTL INPUT	When enabled by $\overline{REN}$, the rising edge of RCLK reads data from the selected queue via the output bus Qout. The queue to be read is selected via the RDADD address bus and a rising edge of RCLK while RADEN is HIGH. A rising edge of RCLK in conjunction with ESTR and RDADD will also select the $\overline{PAE}$n/$\overline{PR}$n flag quadrant to be placed on the $\overline{PAE}$n/$\overline{PR}$n bus during direct flag operation. During polled flag operation the $\overline{PAE}$n/$\overline{PR}$n bus is cycled with respect to RCLK and the ESYNC signal is synchronized to RCLK. The $\overline{PAE}$, $\overline{PR}$ and $\overline{OV}$ outputs are all synchronized to RCLK. During device expansion the EXO and EXI signals are based on RCLK. RCLK must be continuous and free-running.
RDADD [7:0] (RDADD7-P16 RDADD6-P15 RDADD5-P14 RDADD4-N16 RDADD3-N15 RDADD2-N14 RDADD1-M16 RDADD0-M15)	Read Address Bus	HSTL-LVTTL INPUT	For the 32Q device the RDADD bus is 8 bits. The RDADD bus is a dual purpose address bus. The first function of RDADD is to select a Queue to be read from. The least significant 5 bits of the bus, RDADD[4:0] are used to address 1 of 32 possible queues within a multi-queue device. The most significant 3 bits, RDADD[7:5] are used to select 1 of 8 possible multi-queue devices that may be connected in expansion mode. These 3 MSB's will address a device with the matching ID code. The address present on the RDADD bus will be selected on a rising edge of RCLK provided that RADEN is HIGH, (note, that data can be placed on to the Qout bus, read from the previously selected queue on this RCLK edge). On the next rising RCLK edge after a read queue select, a data word from the previous queue will be placed onto the outputs, Qout, regardless of the $\overline{REN}$ input. Two RCLK rising edges after read queue select, data will be placed on to the Qout outputs from the newly selected queue, regardless of $\overline{REN}$ due to the first word fall through effect. The second function of the RDADD bus is to select the quadrant of queues to be loaded on to the $\overline{PAE}$n/$\overline{PR}$n bus during strobed flag mode. The least significant 2 bits, RDADD[1:0] are used to select the quadrant of a device to be placed on the $\overline{PAE}$n bus. The most significant 3 bits, RDADD[7:5] are again used to select 1 of 8 possible multi-queue devices that may be connected in expansion mode. Address bits RDADD[4:2] are don't care during quadrant selection. The quadrant address present on the RDADD bus will be selected on the rising edge of RCLK provided that ESTR is HIGH, (note, that data can be placed on to the Qout bus, read from the previously selected Queue on this RCLK edge). Please refer to Table 2 for details on RDADD bus.

PIN DESCRIPTIONS (CONTINUED)

Symbol & Pin No.	Name	I/O TYPE	Description
$\overline{REN}$ (T11)	Read Enable	HSTL-LVTTL INPUT	The $\overline{REN}$ input enables read operations from a selected Queue based on a rising edge of RCLK. A queue to be read from can be selected via RCLK, RADEN and the RDADD address bus regardless of the state of $\overline{REN}$. Data from a newly selected queue will be available on the Qout output bus on the second RCLK cycle after queue selection regardless of $\overline{REN}$ due to the FWFT operation. A read enable is not required to cycle the $\overline{PAE}$n/$\overline{PR}$n bus (in polled mode) or to select the $\overline{PAE}$n quadrant , (in direct mode).
SCLK (N3)	Serial Clock	HSTL-LVTTL INPUT	If serial programming of the multi-queue device has been selected during master reset, the SCLK input clocks the serial data through the multi-queue device. Data setup on the SI input is loaded into the device on the rising edge of SCLK provided that $\overline{SENI}$ is enabled, LOW. When expansion of devices is performed the SCLK of all devices should be connected to the same source.
$\overline{SENI}$ (M2)	Serial Input Enable	HSTL-LVTTL INPUT	During serial programming of a multi-queue device, data loaded onto the SI input will be clocked into the part (via a rising edge of SCLK), provided the $\overline{SENI}$ input of that device is LOW. If multiple devices are cascaded, the $\overline{SENI}$ input should be connected to the $\overline{SENO}$ output of the previous device. So when serial loading of a given device is complete, its $\overline{SENO}$ output goes LOW, allowing the next device in the chain to be programmed ($\overline{SENO}$ will follow $\overline{SENI}$ of a given device once that device is programmed). The $\overline{SENI}$ input of the master device (or single device), should be controlled by the user.
$\overline{SENO}$ (M1)	Serial Output Enable	HSTL-LVTTL OUTPUT	This output is used to indicate that serial programming or default programming of the multi-queue device has been completed. $\overline{SENO}$ follows $\overline{SENI}$ once programming of a device is complete. Therefore, $\overline{SENO}$ will go LOW after programming provided $\overline{SENI}$ is LOW, once $\overline{SENI}$ is taken HIGH again, $\overline{SENO}$ will also go HIGH. When the $\overline{SENO}$ output goes LOW, the device is ready to begin normal read/write operations. If multiple devices are cascaded and serial programming of the devices will be used, the $\overline{SENO}$ output should be connected to the $\overline{SENI}$ input of the next device in the chain. When serial programming of the first device is complete, $\overline{SENO}$ will go LOW, thereby taking the $\overline{SENI}$ input of the next device LOW and so on throughout the chain. When a given device in the chain is fully programmed the $\overline{SENO}$ output essentially follows the $\overline{SENI}$ input. The user should monitor the $\overline{SENO}$ output of the final device in the chain. When this output goes LOW, serial loading of all devices has been completed.
SI (L1)	Serial In	HSTL-LVTTL INPUT	During serial programming this pin is loaded with the serial data that will configure the multi-queue devices. Data present on SI will be loaded on a rising edge of SCLK provided that $\overline{SENI}$ is LOW. In expansion mode the serial data input is loaded into the first device in a chain. When that device is loaded and its $\overline{SENO}$ has gone LOW, the data present on SI will be directly output to the SO output. The SO pin of the first device connects to the SI pin of the second and so on. The multi-queue device setup registers are shift registers.
SO (M3)	Serial Out	HSTL-LVTTL OUTPUT	This output is used in expansion mode and allows serial data to be passed through devices in the chain to complete programming of all devices. The SI of a device connects to SO of the previous device in the chain. The SO of the final device in a chain should not be connected.
TCK[2] (A8)	JTAG Clock	LVTTL INPUT	Clock input for JTAG function. One of four terminals required by IEEE Standard 1149.1-1990. Test operations of the device are synchronous to TCK. Data from TMS and TDI are sampled on the rising edge of TCK and outputs change on the falling edge of TCK. If the JTAG function is not used this signal needs to be tied to GND.
TDI[2] (B9)	JTAG Test Data Input	LVTTL INPUT	One of four terminals required by IEEE Standard 1149.1-1990. During the JTAG boundary scan operation, test data serially loaded via the TDI on the rising edge of TCK to either the Instruction Register, ID Register and Bypass Register. An internal pull-up resistor forces TDI HIGH if left unconnected.
TDO[2] (A9)	JTAG Test Data Output	LVTTL OUTPUT	One of four terminals required by IEEE Standard 1149.1-1990. During the JTAG boundary scan operation, test data serially loaded output via the TDO on the falling edge of TCK from either the Instruction Register, ID Register and Bypass Register. This output is high impedance except when shifting, while in SHIFT-DR and SHIFT-IR controller states.
TMS[2] (B8)	JTAG Mode Select	LVTTL INPUT	TMS is a serial input pin. One of four terminals required by IEEE Standard 1149.1-1990. TMS directs the device through its TAP controller states. An internal pull-up resistor forces TMS HIGH if left unconnected.
$\overline{TRST}$[2] (C7)	JTAG Reset	LVTTL INPUT	$\overline{TRST}$ is an asynchronous reset pin for the JTAG controller. The JTAG TAP controller does not automatically reset upon power-up, thus it must be reset by either this signal or by setting TMS= HIGH for five TCK cycles. If the TAP controller is not properly reset then the outputs will always be in high-impedance. If the JTAG function is used but the user does not want to use $\overline{TRST}$, then $\overline{TRST}$ can be tied with $\overline{MRS}$ to ensure

PIN DESCRIPTIONS (CONTINUED)

Symbol & Pin No.	Name	I/O TYPE	Description
$\overline{TRST}$[2] (Continued)	JTAG Reset	LVTTL	proper queue operation. If the JTAG function is not used then this signal needs to be tied to GND. An internal pull-up resistor forces $\overline{TRST}$ HIGH if left unconnected.
WADEN (P4)	Write Address Enable	HSTL-LVTTL INPUT	The WADEN input is used in conjunction with WCLK and the WRADD address bus to select a queue to be written in to. A queue addressed via the WRADD bus is selected on the rising edge of WCLK provided that WADEN is HIGH. WADEN should be asserted (HIGH) only during a queue change cycle(s). WADEN should not be permanently tied HIGH. WADEN cannot be HIGH for the same WCLK cycle as FSTR. Note, that a write queue selection cannot be made, (WADEN must NOT go active) until programming of the part has been completed and $\overline{SENO}$ has gone LOW.
WCLK (T7)	Write Clock	HSTL-LVTTL INPUT	When enabled by $\overline{WEN}$, the rising edge of WCLK writes data into the selected Queue via the input bus, Din. The Queue to be written to is selected via the WRADD address bus and a rising edge of WCLK while WADEN is HIGH. A rising edge of WCLK in conjunction with FSTR and WRADD will also select the flag quadrant to be placed on the $\overline{PAF}$n bus during direct flag operation. During polled flag operation the $\overline{PAF}$n bus is cycled with respect to WCLK and the FSYNC signal is synchronized to WCLK. The $\overline{PAF}$n, $\overline{PAF}$ and $\overline{FF}$ outputs are all synchronized to WCLK. During device expansion the FXO and FXI signals are based on WCLK. The WCLK must be continuous and free-running.
$\overline{WEN}$ (T6)	Write Enable	HSTL-LVTTL INPUT	The $\overline{WEN}$ input enables write operations to a selected Queue based on a rising edge of WCLK. A queue to be written to can be selected via WCLK, WADEN and the WRADD address bus regardless of the state of $\overline{WEN}$. Data present on Din can be written to a newly selected queue on the second WCLK cycle after queue selection provided that $\overline{WEN}$ is LOW. A write enable is not required to cycle the $\overline{PAF}$n bus (in polled mode) or to select the $\overline{PAF}$n quadrant , (in direct mode).
WRADD [7:0] (WRADD7-T1 WRADD6-R1 WRADD5-R2 WRADD4-P1 WRADD3-P2 WRADD2-P3 WRADD1-N1 WRADD0-N2)	Write Address Bus	HSTL-LVTTL INPUT	For the 32Q device the WRADD bus is 8 bits. The WRADD bus is a dual purpose address bus. The first function of WRADD is to select a Queue to be written to. The least significant 5 bits of the bus, WRADD[4:0] are used to address 1 of 32 possible queues within a multi-queue device. The most significant 3 bits, WRADD[7:5] are used to select 1 of 8 possible multi-queue devices that may be connected in expansion mode. These 3 MSB's will address a device with the matching ID code. The address present on the WRADD bus will be selected on a rising edge of WCLK provided that WADEN is HIGH, (note, that data present on the Din bus can be written into the previously selected queue on this WCLK edge and on the next rising WCLK also, providing that $\overline{WEN}$ is LOW). Two WCLK rising edges after write queue select, data can be written into the newly selected queue. The second function of the WRADD bus is to select the quadrant of queues to be loaded on to the $\overline{PAF}$n bus during strobed flag mode. The least significant 2 bits, WRADD[1:0] are used to select the quadrant of a device to be placed on the $\overline{PAF}$n bus. The most significant 3 bits, WRADD[7:5] are again used to select 1 of 8 possible multi-queue devices that may be connected in expansion mode. Address bits WRADD[4:2] are don't care during quadrant selection. The quadrant address present on the WRADD bus will be selected on the rising edge of WCLK provided that FSTR is HIGH, (note, that data can be written into the previously selected queue on this WCLK edge). Please refer to Table 1 for details on the WRADD bus.
V_{CC} (See pg. 13)	+2.5V Supply	Power	These are V_{CC} power supply pins and must all be connected to a +2.5V supply rail.
V_{DDQ} (See pg. 13)	O/P Rail Voltage	Power	These pins must be tied to the desired output rail voltage. For LVTTL I/O these pins must be connected to +2.5V, for HSTL these pins must be connected to +1.5V and for eHSTL these pins must be connected to +1.8V.
GND (See pg. 13)	Ground Pin	Ground	These are Ground pins and must all be connected to the GND supply rail.
Vref (K3)	Reference Voltage	HSTL INPUT	This is a Voltage Reference input and must be connected to a voltage level determined from the table "Recommended DC Operating Conditions". The input provides the reference level for HSTL/eHSTL inputs. For LVTTL I/O mode this input should be tied to GND.

NOTES:
1. Inputs should not change after Master Reset.
2. These pins are for the JTAG port. Please refer to pages 59-62 and Figures 36-38.

PIN NUMBER TABLE

Symbol	Name	I/O TYPE	Pin Number
D[35:0] Din	Data Input Bus	HSTL-LVTTL INPUT	D35-J3, D(34-32)-H(3-1), D(31-29)-G(3-1), D(28-26)-F(3-1), D(25-23)-E(3-1), D(22-20)-D(3-1), D(19-17)-C(3-1), D(16,15)-B(2,1), D(14-12)-A(1-3), D11-B3, D10-A4, D9-B4, D8-C4, D7-A5, D6-B5, D5-C5, D4-A6, D3-B6, D2-C6, D1-A7, D0-B7
Q[35:0] Qout	Data Output Bus	HSTL-LVTTL OUTPUT	Q(35,34)-J(15,16), Q(33-31)-H(14-16), Q(30-28)-G(14-16), Q(27-25)-F(14-16), Q(24-22)-E(14-16), Q(21,20)-D(15,16), Q19-B16, Q(18,17)-C(16,15), Q16-D14, Q(15,14)-A(16,15), Q13-B15, Q12-A14, Q11-B14, Q10-C14, Q9-A13, Q8-B13, Q7-C13, Q6-A12, Q5-B12, Q4-C12, Q3-A11, Q2-B11, Q(1,0)-C(11,10)
V_{CC}	+2.5V Supply	Power	D(7-10), E(6,7,10,11), F(5,12), G(4,5,12,13), H(4,13), J(4,13), K(4,5,12,13), L(5,12), M(6,7,10,11), N(7-10)
V_{DDQ}	O/P Rail Voltage	Power	D(4-6,11-13), E(4,5,12,13), F(4,13), L(4,13), M(4,5,12,13), N(4-6,11-13)
GND	Ground Pin	Ground	E(8-9), F(6-11), G(6-11), H(5-12), J(1,5-12), K(2,6-11,14), L(6-11), M(8-9)

ABSOLUTE MAXIMUM RATINGS

Symbol	Rating	Commercial	Unit
V_{TERM}	Terminal Voltage with respect to GND	–0.5 to +3.6(2)	V
T_{STG}	Storage Temperature	–55 to +125	°C
I_{OUT}	DC Output Current	–50 to +50	mA

NOTES:

1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
2. Compliant with JEDEC JESD8-5. V_{CC} terminal only.

CAPACITANCE (T_A = +25°C, f = 1.0MHz)

Symbol	Parameter(1)	Conditions	Max.	Unit
C_{IN}(2,3)	Input Capacitance	V_{IN} = 0V	10(3)	pF
C_{OUT}(1,2)	Output Capacitance	V_{OUT} = 0V	15	pF

NOTES:

1. With output deselected, ($\overline{OE} \geq V_{IH}$).
2. Characterized values, not currently tested.
3. C_{IN} for Vref is 20pF.

RECOMMENDED DC OPERATING CONDITIONS

Symbol	Parameter		Min.	Typ.	Max.	Unit
V_{CC}	Supply Voltage		2.375	2.5	2.625	V
GND	Supply Voltage		0	0	0	V
V_{IH}	Input High Voltage	— LVTTL	1.7	—	3.45	V
		— eHSTL	V_{REF}+0.2	—	—	V
		— HSTL	V_{REF}+0.2	—	—	V
V_{IL}	Input Low Voltage	— LVTTL	-0.3	—	0.7	V
		— eHSTL	—	—	V_{REF}-0.2	V
		— HSTL	—	—	V_{REF}-0.2	V
V_{REF} (HSTL only)	Voltage Reference Input	— eHSTL	0.8	0.9	1.0	V
		— HSTL	0.68	0.75	0.9	V
T_A	Operating Temperature Commercial		0	—	70	°C
T_A	Operating Temperature Industrial		-40	—	85	°C

NOTE:

1. V_{REF} is only required for HSTL or eHSTL inputs. V_{REF} should be tied LOW for LVTTL operation.

DC ELECTRICAL CHARACTERISTICS

(Commercial: V_{CC} = 2.5V ± 0.125V, T_A = 0°C to +70°C;Industrial: V_{CC} = 2.5V ± 0.125V, T_A = -40°C to +85°C)

Symbol	Parameter		Min.	Max.	Unit
I_{LI}	Input Leakage Current		–10	10	µA
I_{LO}	Output Leakage Current		–10	10	µA
V_{OH}[3]	Output Logic "1" Voltage,	I_{OH} = –8 mA @V_{DDQ} = 2.5V ± 0.125V (LVTTL)	V_{DDQ}-0.4	—	V
		I_{OH} = –8 mA @V_{DDQ} = 1.8V ± 0.1V (eHSTL)	V_{DDQ}-0.4	—	V
		I_{OH} = –8 mA @V_{DDQ} = 1.5V ± 0.1V (HSTL)	V_{DDQ}-0.4	—	V
V_{OL}	Output Logic "0" Voltage,	I_{OL} = 8 mA @V_{DDQ} = 2.5V ± 0.125V (LVTTL)	—	0.4V	V
		I_{OL} = 8 mA @V_{DDQ} = 1.8V ± 0.1V (eHSTL)	—	0.4V	V
		I_{OL} = 8 mA @V_{DDQ} = 1.5V ± 0.1V (HSTL)	—	0.4V	V
I_{CC1}[1,2]	Active V_{CC} Current (V_{CC} = 2.5V)	I/O = LVTTL	—	80	mA
		I/O = HSTL	—	150	mA
		I/O = eHSTL	—	150	mA
I_{CC2}[1]	Standby V_{CC} Current (V_{CC} = 2.5V)	I/O = LVTTL	—	25	mA
		I/O = HSTL	—	100	mA
		I/O = eHSTL	—	100	mA
I_{CC3}[1]	Standby V_{CC} Current in Power Down mode(V_{CC} = 2.5V)	I/O = LVTTL	—	—	mA
		I/O = HSTL	—	50	mA
		I/O = eHSTL	—	50	mA
I_{DDQ}[1,2]	Active V_{DDQ} Current (V_{DDQ} = 2.5V LVTTL)	I/O = LVTTL	—	10	mA
	(V_{DDQ} = 1.5V HSTL)	I/O = HSTL	—	10	mA
	(V_{DDQ} = 1.8V eHSTL)	I/O = eHSTL	—	10	mA

NOTES:

1. Both WCLK and RCLK toggling at 20MHz.
2. Data inputs toggling at 10MHz.
3. Total Power consumed: PT = [(V_{CC} x I_{CC}) + (V_{DDQ} x I_{DDQ})].
4. Outputs are not 3.3V tolerant.
5. The following inputs should be pulled to GND: WRADD, RDADD, WADEN, FSTR, ESTR, SCLK, SI, EXI, FXI and all Data Inputs. The following inputs should be pulled to V_{CC}: $\overline{WEN}$, $\overline{REN}$, $\overline{SENI}$, $\overline{PRS}$, $\overline{MRS}$, TDI, TMS and $\overline{TRST}$. All other inputs are don't care and should be at a known state.

HSTL
1.5V AC TEST CONDITIONS

Input Pulse Levels	0.25 to 1.25V
Input Rise/Fall Times	0.4ns
Input Timing Reference Levels	0.75
Output Reference Levels	VDDQ/2

NOTE:
1. VDDQ = 1.5V±.

EXTENDED HSTL
1.8V AC TEST CONDITIONS

Input Pulse Levels	0.4 to 1.4V
Input Rise/Fall Times	0.4ns
Input Timing Reference Levels	0.9
Output Reference Levels	VDDQ/2

NOTE:
1. VDDQ = 1.8V±.

2.5V LVTTL
2.5V AC TEST CONDITIONS

Input Pulse Levels	GND to 2.5V
Input Rise/Fall Times	1ns
Input Timing Reference Levels	VCC/2
Output Reference Levels	VDDQ/2

NOTE:
1. For LVTTL VCC = VDDQ.

AC TEST LOADS

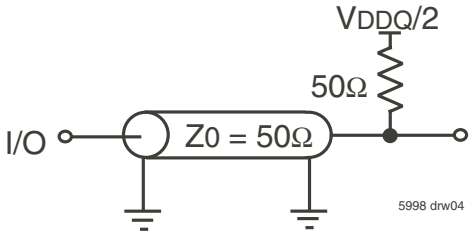


Figure 2a. AC Test Load

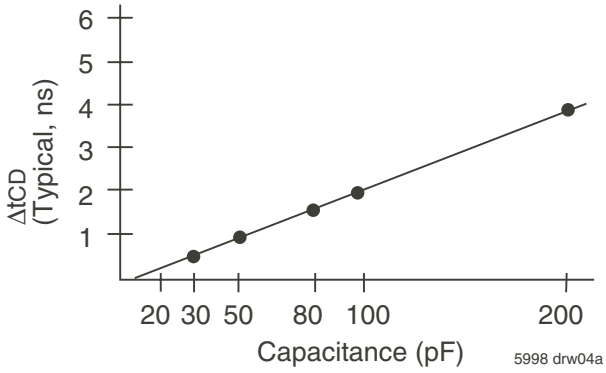
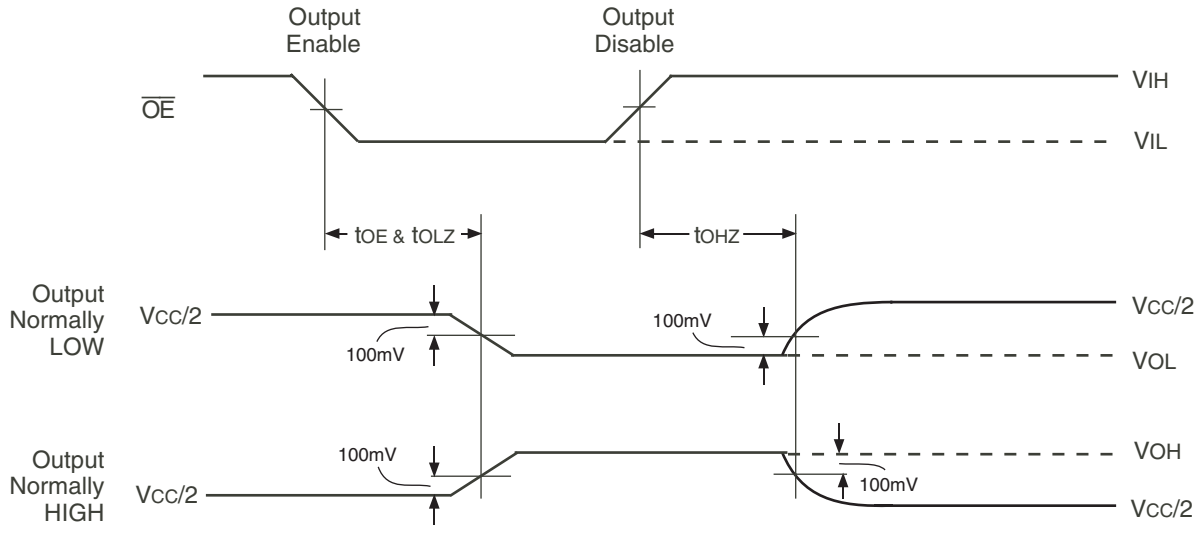


Figure 2b. Lumped Capacitive Load, Typical Derating

OUTPUT ENABLE & DISABLE TIMING



NOTE:
1. $\overline{REN}$ is HIGH.

AC ELECTRICAL CHARACTERISTICS

(Commercial: V_{CC} = 2.5V ± 0.15V, T_A = 0°C to +70°C;Industrial: V_{CC} = 2.5V ± 0.15V, T_A = -40°C to +85°C; JEDEC JESD8-A compliant)

Symbol	Parameter	Commercial IDT72T51546L5 IDT72T51556L5		Com'l & Ind'l[1] IDT72T51546L6 IDT72T51556L6		Unit
		Min.	Max.	Min.	Max.	
f_S	Clock Cycle Frequency (WCLK & RCLK)	—	200	—	166	MHz
t_A	Data Access Time	0.6	3.6	0.6	3.7	ns
t_{CLK}	Clock Cycle Time	5	—	6	—	ns
t_{CLKH}	Clock High Time	2.3	—	2.7	—	ns
t_{CLKL}	Clock Low Time	2.3	—	2.7	—	ns
t_{DS}	Data Setup Time	1.5	—	2.0	—	ns
t_{DH}	Data Hold Time	0.5	—	0.5	—	ns
t_{ENS}	Enable Setup Time	1.5	—	2.0	—	ns
t_{ENH}	Enable Hold Time	0.5	—	0.5	—	ns
t_{RS}	Reset Pulse Width	30	—	30	—	ns
t_{RSS}	Reset Setup Time	15	—	15	—	ns
t_{RSR}	Reset Recovery Time	10	—	10	—	ns
t_{PRSS}	Partial Reset Setup	1.5	—	2.0	—	ns
t_{PRSH}	Partial Reset Hold	0.5	—	0.5	—	ns
t_{OLZ} ($\overline{OE}$-Q_n)[2]	Output Enable to Output in Low-Impedance	0.6	3.6	0.6	3.7	ns
t_{OHZ}[2]	Output Enable to Output in High-Impedance	0.6	3.6	0.6	3.7	ns
t_{OE}	Output Enable to Data Output Valid	0.6	3.6	0.6	3.7	ns
f_C	Clock Cycle Frequency (SCLK)	—	10	—	10	MHz
t_{SCLK}	Serial Clock Cycle	100	—	100	—	ns
t_{SCKH}	Serial Clock High	45	—	45	—	ns
t_{SCKL}	Serial Clock Low	45	—	45	—	ns
t_{SDS}	Serial Data In Setup	20	—	20	—	ns
t_{SDH}	Serial Data In Hold	1.2	—	1.2	—	ns
t_{SENS}	Serial Enable Setup	20	—	20	—	ns
t_{SENH}	Serial Enable Hold	1.2	—	1.2	—	ns
t_{SDO}	SCLK to Serial Data Out	—	20	—	20	ns
t_{SENO}	SCLK to Serial Enable Out	—	20	—	20	ns
t_{SDOP}	Serial Data Out Propagation Delay	1.5	3.7	1.5	3.7	ns
t_{SENOP}	Serial Enable Propagation Delay	1.5	3.7	1.5	3.7	ns
t_{PCWQ}	Programming Complete to Write Queue Selection	20	—	20	—	ns
t_{PCRQ}	Programming Complete to Read Queue Selection	20	—	20	—	ns
t_{AS}	Address Setup	1.5	—	2.5	—	ns
t_{AH}	Address Hold	1.0	—	1.5	—	ns
t_{WFF}	Write Clock to Full Flag	—	3.6	—	3.7	ns
t_{ROV}	Read Clock to Output Valid	—	3.6	—	3.7	ns
t_{STS}	$\overline{PAE}$/$\overline{PAF}$ Strobe Setup	1.5	—	2.0	—	ns
t_{STH}	$\overline{PAE}$/$\overline{PAF}$ Strobe Hold	0.5	—	0.5	—	ns
t_{QS}	Queue Setup	1.5	—	2.0	—	ns
t_{QH}	Queue Hold	1.0	—	0.5	—	ns
t_{WAF}	WCLK to $\overline{PAF}$ flag	0.6	3.6	0.6	3.7	ns
t_{RAE}	RCLK to $\overline{PAE}$ flag	0.6	3.6	0.6	3.7	ns
t_{PAF}	Write Clock to Synchronous Almost-Full Flag Bus	0.6	3.6	0.6	3.7	ns
t_{PAE}	Read Clock to Synchronous Almost-Empty Flag Bus	0.6	3.6	0.6	3.7	ns

NOTES:
1. Industrial temperature range product for the 6ns is available as a standard device. All other speed grades available by special order.
2. Values guaranteed by design, not currently tested.

AC ELECTRICAL CHARACTERISTICS (CONTINUED)

(Commercial: V_{CC} = 2.5V ± 0.15V, T_A = 0°C to +70°C;Industrial: V_{CC} = 2.5V ± 0.15V, T_A = -40°C to +85°C; JEDEC JESD8-A compliant)

Symbol	Parameter	Commercial IDT72T51546L5 IDT72T51556L5		Com'l & Ind'l[1] IDT72T51546L6 IDT72T51556L6		Unit
		Min.	Max.	Min.	Max.	
t_{ERCLK}	RCLK to Echo RCLK Output	—	4.0	—	4.2	ns
t_{CLKEN}	RCLK to Echo $\overline{REN}$ Output	—	3.6	—	3.7	ns
t_{PAELZ}[2]	RCLK to $\overline{PAE}$ Flag Bus to Low-Impedance	0.6	3.6	0.6	3.7	ns
t_{PAEHZ}[2]	RCLK to $\overline{PAE}$ Flag Bus to High-Impedance	0.6	3.6	0.6	3.7	ns
t_{PAFLZ}[2]	WCLK to $\overline{PAF}$ Flag Bus to Low-Impedance	0.6	3.6	0.6	3.7	ns
t_{PAFHZ}[2]	WCLK to $\overline{PAF}$ Flag Bus to High-Impedance	0.6	3.6	0.6	3.7	ns
t_{FFHZ}[2]	WCLK to Full Flag to High-Impedance	0.6	3.6	0.6	3.7	ns
t_{FFLZ}[2]	WCLK to Full Flag to Low-Impedance	0.6	3.6	0.6	3.7	ns
t_{OVLZ}[2]	RCLK to Output Valid Flag to Low-Impedance	0.6	3.6	0.6	3.7	ns
t_{OVHZ}[2]	RCLK to Output Valid Flag to High-Impedance	0.6	3.6	0.6	3.7	ns
t_{FSYNC}	WCLK to $\overline{PAF}$ Bus Sync to Output	0.6	3.6	0.6	3.7	ns
t_{FXO}	WCLK to $\overline{PAF}$ Bus Expansion to Output	0.6	3.6	0.6	3.7	ns
t_{ESYNC}	RCLK to $\overline{PAE}$ Bus Sync to Output	0.6	3.6	0.6	3.7	ns
t_{EXO}	RCLK to $\overline{PAE}$ Bus Expansion to Output	0.6	3.6	0.6	3.7	ns
t_{PR}	RCLK to Packet Ready Flag	0.6	3.6	0.6	3.7	ns
t_{SKEW1}	SKEW time between RCLK and WCLK for $\overline{FF}$ and $\overline{OV}$	4	—	4.5	—	ns
t_{SKEW2}	SKEW time between RCLK and WCLK for $\overline{PAF}$ and $\overline{PAE}$	5	—	6	—	ns
t_{SKEW3}	SKEW time between RCLK and WCLK for $\overline{PAF}$[0:7] and $\overline{PAE}$[0:7]	5	—	6	—	ns
t_{SKEW4}	SKEW time between RCLK and WCLK for $\overline{PR}$ and $\overline{OV}$	5	—	6	—	ns
t_{SKEW5}	SKEW time between RCLK and WCLK for $\overline{OV}$ when in Packet Mode	8	—	10	—	ns
t_{XIS}	Expansion Input Setup	1.0	—	1.0	—	ns
t_{XIH}	Expansion Input Hold	0.5	—	0.5	—	ns

NOTES:
1. Industrial temperature range product for the 6ns is available as a standard device. All other speed grades available by special order.
2. Values guaranteed by design, not currently tested.

FUNCTIONAL DESCRIPTION

MASTER RESET

A Master Reset is performed by toggling the $\overline{MRS}$ input from HIGH to LOW to HIGH. During a master reset all internal multi-queue device setup and control registers are initialized and require programming either serially by the user via the serial port, or using the default settings. During a master reset the state of the following inputs determine the functionality of the part, these pins should be held HIGH or LOW.

PKT – Packet Mode
FM – Flag bus Mode
IW, OW, BM – Bus Matching options
MAST – Master Device
ID0, 1, 2 – Device ID
DFM – Programming mode, serial or default
DF – Offset value for $\overline{PAE}$ and $\overline{PAF}$

Once a master reset has taken place, the device must be programmed either serially or via the default method before any read/write operations can begin.

See Figure 5, *Master Reset* for relevant timing.

PARTIAL RESET

A Partial Reset is a means by which the user can reset both the read and write pointers of a single queue that has been setup within a multi-queue device. Before a partial reset can take place on a queue, the respective queue must be selected on both the read port and write port a minimum of 2 RCLK and 2 WCLK cycles before the $\overline{PRS}$ goes LOW. The partial reset is then performed by toggling the $\overline{PRS}$ input from HIGH to LOW to HIGH, maintaining the LOW state for at least one WCLK and one RCLK cycle. Once a partial reset has taken place a minimum of 3 WCLK and 3 RCLK cycles must occur before enabled writes or reads can occur.

A Partial Reset only resets the read and write pointers of a given queue, a partial reset will not effect the overall configuration and setup of the multi-queue device and its queues.

See Figure 6, *Partial Reset* for relevant timing.

SERIAL PROGRAMMING

The multi-queue flow-control device is a fully programmable device, providing the user with flexibility in how queues are configured in terms of the number of queues, depth of each queue and position of the $\overline{PAF}$/$\overline{PAE}$ flags within respective queues. All user programming is done via the serial port after a master reset has taken place. Internally the multi-queue device has setup registers which must be serially loaded, these registers contain values for every queue within the device, such as the depth and $\overline{PAE}$/$\overline{PAF}$ offset values. The IDT72T51546/72T51556 devices are capable of up to 32 queues and therefore contain 32 sets of registers for the setup of each queue.

During a Master Reset if the DFM (Default Mode) input is LOW, then the device will require serial programming by the user. It is recommended that the user utilize a 'C' program provided by IDT, this program will prompt the user for all information regarding the multi-queue setup. The program will then generate a serial bit stream which should be serially loaded into the device via the serial port. For the IDT72T51546/72T51556 devices the serial programming requires a total number of serially loaded bits per device, (SCLK cycles with $\overline{SENI}$ enabled), calculated by: 19+(Qx72) where Q is the number of queues the user wishes to setup within the device.

Once the master reset is complete and $\overline{MRS}$ is HIGH, the device can be serially loaded. Data present on the SI (serial in), input is loaded into the serial port on a rising edge of SCLK (serial clock), provided that $\overline{SENI}$ (serial in enable), is LOW. Once serial programming of the device has been successfully completed the device will indicate this via the $\overline{SENO}$ (serial output enable) going active, LOW. Upon detection of completion of programming, the user should cease all programming and take $\overline{SENI}$ inactive, HIGH. Note, $\overline{SENO}$ follows $\overline{SENI}$ once programming of a device is complete. Therefore, $\overline{SENO}$ will go LOW after programming provided $\overline{SENI}$ is LOW, once $\overline{SENI}$ is taken HIGH again, $\overline{SENO}$ will also go HIGH. The operation of the SO output is similar, when programming of a given device is complete, the SO output will follow the SI input.

If devices are being used in expansion mode the serial ports of devices should be cascaded. The user can load all devices via the serial input port control pins, SI & $\overline{SENI}$, of the first device in the chain. Again, the user may utilize the 'C' program to generate the serial bit stream, the program prompting the user for the number of devices to be programmed. The $\overline{SENO}$ and SO (serial out) of the first device should be connected to the $\overline{SENI}$ and SI inputs of the second device respectively and so on, with the $\overline{SENO}$ & SO outputs connecting to the $\overline{SENI}$ & SI inputs of all devices through the chain. All devices in the chain should be connected to a common SCLK. The serial output port of the final device should be monitored by the user. When $\overline{SENO}$ of the final device goes LOW, this indicates that serial programming of all devices has been successfully completed. Upon detection of completion of programming, the user should cease all programming and take $\overline{SENI}$ of the first device in the chain inactive, HIGH.

As mentioned, the first device in the chain has its serial input port controlled by the user, this is the first device to have its internal registers serially loaded by the serial bit stream. When programming of this device is complete it will take its $\overline{SENO}$ output LOW and bypass the serial data loaded on the SI input to its SO output. The serial input of the second device in the chain is now loaded with the data from the SO of the first device, while the second device has its $\overline{SENI}$ input LOW. This process continues through the chain until all devices are programmed and the $\overline{SENO}$ of the final device goes LOW.

Once all serial programming has been successfully completed, normal operations, (queue selections on the read and write ports) may begin. When connected in expansion mode, the IDT72T51546/72T51556 devices require a total number of serially loaded bits per device to complete serial programming, (SCLK cycles with $\overline{SENI}$ enabled), calculated by: n[19+(Qx72)] where Q is the number of queues the user wishes to setup within the device, where n is the number of devices in the chain.

See Figure 7, *Serial Port Connection* and Figure 8, *Serial Programming* for connection and timing information.

DEFAULT PROGRAMMING

During a Master Reset if the DFM (Default Mode) input is HIGH the multi-queue device will be configured for default programming, (serial programming is not permitted). Default programming provides the user with a simpler, however limited means by which to setup the multi-queue flow-control device, rather than using the serial programming method. The default mode will configure a multi-queue device such that the maximum number of queues possible are setup, with all of the parts available memory blocks being allocated equally between the queues. The values of the $\overline{PAE}$/$\overline{PAF}$ offsets is determined by the state of the DF (default) pin during a master reset.

For the IDT72T51546/72T51556 devices the default mode will setup 32 queues, each queue being 1024 x36 and 2048 x36 deep respectively. For both devices the value of the $\overline{PAE}$/$\overline{PAF}$ offsets is determined at master reset by the state of the DF input. If DF is LOW then both the $\overline{PAE}$ & $\overline{PAF}$ offset will be 8, if HIGH then the value is 128.

When configuring the IDT72T51546/72T51556 devices in default mode the user simply has to apply WCLK cycles after a master reset, until $\overline{SENO}$ goes LOW, this signals that default programming is complete. These clock cycles are required for the device to load its internal setup registers. When a single multi-queue device is used, the completion of device programming is signaled by the

$\overline{SENO}$ output of a device going from HIGH to LOW. Note, that $\overline{SENI}$ must be held LOW when a device is setup for default programming mode.

When multi-queue devices are connected in expansion mode, the $\overline{SENI}$ of the first device in a chain can be held LOW. The $\overline{SENO}$ of a device should connect to the $\overline{SENI}$ of the next device in the chain. The $\overline{SENO}$ of the final device is used to indicate that default programming of all devices is complete. When the final $\overline{SENO}$ goes LOW normal operations may begin. Again, all devices will be programmed with their maximum number of queues and the memory divided equally between them. Please refer to Figure 9, *Default Programming*.

READING AND WRITING TO THE IDT MULTI-QUEUE FLOW-CONTROL DEVICE

The IDT72T51546/72T51556 multi-queue flow-control devices can be configured in two distinct modes, namely Standard Mode and Packet Mode.

STANDARD MODE OPERATION (PKT = LOW ON MASTER RESET)

WRITE QUEUE SELECTION AND WRITE OPERATION (STANDARD MODE)

The IDT72T51546/72T51556 multi-queue flow-control devices can be configured up to a maximum of 32 queues into which data can be written via a common write port using the data inputs (Din), write clock (WCLK) and write enable ($\overline{WEN}$). The queue to be written is selected by the address present on the write address bus (WRADD) during a rising edge on WCLK while write address enable (WADEN) is HIGH. The state of $\overline{WEN}$ does not impact the queue selection. The queue selection requires 1 WCLK cycle. All subsequent data writes will be to this queue until another queue is selected.

Standard mode operation is defined as individual words will be written to the device as opposed to Packet Mode where complete packets may be written. The write port is designed such that 100% bus utilization can be obtained. This means that data can be written into the device on every WCLK rising edge including the cycle that a new queue is being addressed.

Changing queues requires a minimum of 3 WCLK cycles on the write port (see Figure 10, *Write Queue Select, Write Operation and Full flag Operation*). WADEN goes high signaling a change of queue (clock cycle "A"). The address on WRADD at that time determines the next queue. Data presented during that cycle ("A") and the next cycle ("B" and "C"), will be written to the active (old) queue, provided $\overline{WEN}$ is active LOW. If $\overline{WEN}$ is HIGH (inactive) for these 3 clock cycles, data will not be written in to the previous queue. The write port discrete full flag will update to show the full status of the newly selected queue (Q_X) at this last cycle's rising edge ("C"). Data present on the data input bus (Din), can be written into the newly selected queue (Q_X) on the rising edge of WCLK on the third cycle ("D") following a change of queue, provided $\overline{WEN}$ is LOW and the new queue is not full. If the newly selected queue is full at the point of its selection, any writes to that queue will be prevented. Data cannot be written into a full queue.

Refer to Figure 10, *Write Queue Select, Write Operation and Full flag Operation*, Figure 11, *Write Operations & First Word Fall Through* for timing diagrams and Figure 12, *Full Flag Timing in Expansion Mode* for timing diagrams.

TABLE 1 — WRITE ADDRESS BUS, WRADD[7:0]

Operation	WCLK	WADEN	FSTR	WRADD[7:0]		
Write Queue Select	↑	1	0	7 6 5 Device Select (Compared to ID0,1,2)	4 3 2 Write Queue Address (5 bits = 32 Queues)	1 0
$\overline{PAFn}$ Quadrant Select	↑	0	1	7 6 5 Device Select (Compared to ID0,1,2)	4 3 2 X X X	1 0 Quadrant Address

Quadrant Address	Queue Status on $\overline{PAFn}$ Bus
00	Q0 : Q7 → $\overline{PAF0}$: $\overline{PAF7}$
01	Q8 : Q15 → $\overline{PAF0}$: $\overline{PAF7}$
10	Q16 : Q23 → $\overline{PAF0}$: $\overline{PAF7}$
11	Q24 : Q31 → $\overline{PAF0}$: $\overline{PAF7}$

5998 drw05

READ QUEUE SELECTION AND READ OPERATION (STANDARD MODE)

The IDT72T51546/72T51556 multi-queue flow-control devices can be configured up to a maximum of 32 queues which data can be read via a common read port using the data outputs (Qout), read clock (RCLK) and read enable ($\overline{REN}$). An output enable, $\overline{OE}$ control pin is also provided to allow High-Impedance selection of the Qout data outputs. The multi-queue device read port operates in a mode similar to "First Word Fall Through" on a SuperSync IDT FIFO, but with the added feature of data output pipelining (see Figure 11, *Write Operations & First Word Fall Through*). The queue to be read is selected by the address presented on the read address bus (RDADD) during a rising edge on RCLK while read address enable (RADEN) is HIGH. The state of $\overline{REN}$ does not impact the queue selection. The queue selection requires 1 RCLK cycles. All subsequent data reads will be from this queue until another queue is selected.

Standard mode operation is defined as individual words will be read from the device as opposed to Packet Mode where complete packets may be read. The read port is designed such that 100% bus utilization can be obtained. This means that data can be read out of the device on every RCLK rising edge including the cycle that a new queue is being addressed.

Changing queues requires a minimum of three RCLK cycles on the read port (see Figure 13, *Read Queue Select, Read Operation*). RADEN goes high signaling a change of queue (clock cycle "D"). The address on RDADD at that time determines the next queue. Data presented during that cycle ("D") will be read at "D" ($+t_A$), and the next cycle ("E"), can continue to be read from the active (old) queue (Q_P), provided $\overline{REN}$ is active LOW. If $\overline{REN}$ is HIGH (inactive) for these two clock cycles, data will not be read from the previous queue. The next cycle's rising edge ("F"), the read port discrete empty flag will update to show the empty status of the newly selected queue (Q_F). The internal pipeline is also loaded at this time ("F") with the last word from the previous (old) queue (Q_F) as well as the next word from the new queue (Q_F). Both of these words will fall through to the output register (provided the $\overline{OE}$ is asserted) consecutively (cycles "F" and "G" respectively) following the selection of the new queue regardless of the state of $\overline{REN}$, unless the new queue (Q_F) is empty. If the newly selected queue is empty, any reads from that queue will be prevented. Data cannot be read from an empty queue. The last word in the data output register (from the previous queue), will remain on the data bus, but the output valid flag, $\overline{OV}$ will go HIGH, to indicate that the data present is no longer valid. This pipelining effect provides the user with 100% bus utilization, and brings about the possibility that a "NULL" queue may be required within a multi-queue device. Null queue operation is discussed in the next section. Remember that $\overline{OE}$ allows the user to place the data output bus (Qout) into High-Impedance and the data can be read in to the output register regardless of $\overline{OE}$.

Refer to Table 2, for Read Address Bus arrangement. Also, refer to Figures 13, 15, and 16 for read queue selection and read port operation timing diagrams.

PACKET MODE OPERATION (PKT = HIGH on Master Reset)

The Packet mode operation provides the capability where, user defined packets or frames can be written to the device as opposed to Standard mode where individual words are written. For clarification, in Packet Mode, a packet can be written to the device with the starting location designated as Transmit Start of Packet (TSOP) and the ending location designated as Transmit End of Packet (TEOP). In conjunction, a packet read from the device will be designated as

TABLE 2 — READ ADDRESS BUS, RDADD[7:0]

Operation	RCLK	RADEN	ESTR	Null-Q	RDADD[7:0]		
Read Queue Select	↑	1	0	0	7 6 5 Device Select (Compared to ID0,1,2)	4 3 2 1 0 Read Queue Address (5 bits = 32 Queues)	
$\overline{PAE}$n/$\overline{PR}$n Quadrant Select	↑	0	1	0	7 6 5 Device Select (Compared to ID0,1,2)	4 3 2 X X X	1 0 Quadrant Address
Null Queue Select	↑	1	0	1	7 6 5 X X X	4 3 2 X X X	1 0 X X

Quadrant Address	Queue Status on $\overline{PAE}$n/$\overline{PR}$n Bus
00	Q0 : Q7 → $\overline{PAE}$0 : $\overline{PAE}$7
01	Q8 : Q15 → $\overline{PAE}$0 : $\overline{PAE}$7
10	Q16 : Q23 → $\overline{PAE}$0 : $\overline{PAE}$7
11	Q24 : Q31 → $\overline{PAE}$0 : $\overline{PAE}$7

5998 drw06

Receive Start of Packet (RSOP) and a Receive End of Packet (REOP). The minimum size for a packet is four words (SOP, two words of data and EOP). The almost empty flag bus becomes the "Packet Ready" $\overline{PR}$ flag bus when the device is configured for packet mode. Valid packets are indicated when both $\overline{PR}$ and $\overline{OV}$ are asserted.

WRITE QUEUE SELECTION AND WRITE OPERATION (PACKET MODE)

It is required that a full packet be written to a queue before moving to a different queue. The device requires three cycles to change queues. Packet mode, has 2 restrictions: <1> An extra word (or filler word) is required to be written after each packet on the cycle following the queue change to ensure the RSOP in the old queue is not read out on a queue change because of the first word fall through. <2> No SOP/EOP is allowed to read/written at cycle ("D" or "K") the second cycle after a queue change. In this mode, the write port may not obtain 100% bus utilization.

Changing queues requires a minimum of 3 WCLK cycles on the write port (see Figure 17, *Writing in Packet Mode during a Queue Change*). WADEN goes high signaling a change of queue (clock cycle "B" or "I"). The address on WRADD at the rising edge of WCLK determines the next queue. Data presented on Din during that cycle ("B" or "I") and the next cycle ("C" or "J") can continue to be written to the active (old) queue (Q_A or Q_B respectively), provided $\overline{WEN}$ is LOW (active). If $\overline{WEN}$ is HIGH (inactive) for these two clock cycles (H), data will not be written in to the previous queue (Q_A). The second cycle following a request for queue change ("D" or "K") will require a "filler" word to be written to the device. This can be done by clocking the TEOP twice or by writing a "filler" word. In packet mode, the multi-queue is designed under the 2 restrictions listed previously. Note, an erroneous Packet Ready flag may occur if the EOP or SOP marker shows up at the second cycle after a queue change. To prevent an erroneous Packet Ready flag from occurring a filler word should be written into the old queue at the last clock cycle of writing. It is important to know that no SOP or EOP may be written into the device during this cycle ("D" or "K"). The write port discrete full flag will update to show the full status of the newly selected queue (Q_B) at this last cycle's rising edge ("D" or "K"). Data values presented on the data input bus (Din), can be written into the newly selected queue (Q_X) on the rising edge of WCLK on the third cycle ("E") following a request for change of queue, provided $\overline{WEN}$ is LOW (active) and the new queue is not full. If a selected queue is full ($\overline{FF}$ is LOW), then writes to that queue will be prevented. Note, data cannot be written into a full queue.

Refer to Figure 17, *Writing in Packet Mode during a Queue Change* and Figure 19, *Data Input (Transit) Packet Mode of Operation* for timing diagrams.

READ QUEUE SELECTION AND READ OPERATION (PACKET MODE)

In packet Mode it is required that a full packet is read from a queue before moving to a different queue. The device requires three cycles to change queues. In Packet Mode, there are 2 restrictions <1> An extra word (or filler word) should have been inserted into the data stream after each packet to insure the RSOP in the old queue is not read out on a queue change because of the first word fall through and this word should be discarded. <2> No EOP/SOP is allowed to be read/written at cycle ("D" or "K") the second cycle after a queue change). In this mode, the read port may not obtain 100% bus utilization.

Changing queues requires a minimum of 3 RCLK cycles on the read port (see Figure 18, *Reading in Packet Mode during a Queue Change*). RADEN goes high signaling a change of queue (clock cycle "B" or "I"). The address on RDADD at the rising edge of RCLK determines the queue. As illustrated in Figure 18 during cycle ("B" or "I"), and the next cycle ("C" or "J") data can continue to be read from the active (old) queue (Q_A or Q_B respectively), provided both $\overline{REN}$ and $\overline{OE}$ are LOW (active) simultaneously with changing queues. REOP for packet located in queue (Q_A) must be read on or before a queue change request is made ("C" or "J"). If $\overline{REN}$ is HIGH (inactive) for these two clock cycles, data will not be read from the previous queue (Q_A). In applications where the multi-queue flow-control device is connected to a shared bus, an output enable, $\overline{OE}$ control pin is also provided to allow High-Impedance selection of the data outputs (Qout). With reference to Figure 18 when changing queues, a packet marker (SOP or EOP) should not be read on cycle ("E" or "L"). Reading a SOP or EOP should not occur during the cycles required for a queue change. It is also recommended that a queue change should not occur once the reading of the packet has commenced, The EOP marker of the packet prior to a queue change should be read on or before the queue change. If the EOP word is read before a queue change, $\overline{REN}$ can be pulled high to disable further reads. When the queue change is initiated, the filler word written into the current queue after the EOP word will fall through followed by and the first word from the new queue.

Refer to Figure 18, *Reading in Packet Mode during a Queue Change* as well as Figures 13, 15, and 16 for timing diagrams and Table 2, for Read Address bus arrangement.

Note, the almost empty flag bus becomes the "Packet Ready" flag bus when the device is configured for packet ready mode. .

PACKET READY FLAG

The 36-bit multi-queue flow-control device provides the user with a Packet Ready feature. During a Master Reset the logic "1" (HIGH) on the PKT input signal (packet mode select), configures the device in packet mode. The $\overline{PR}$ discrete flag, provides a packet ready status of the active queue selected on the read port. A packet ready status is individually maintained on all queues; however only the queue selected on the read port has its packet ready status indicated on the $\overline{PR}$ output flag. A packet is available on the output for reading when both $\overline{PR}$ and $\overline{OV}$ are asserted LOW. If less than a full packet is available, the $\overline{PR}$ flag will be HIGH (packet not ready). In packet mode, no words can be read from a queue until a complete packet has been written into that queue, regardless of $\overline{REN}$.

When packet mode is selected the Programmable Almost Empty bus, $\overline{PAE}$n, becomes the Packet Ready bus, $\overline{PR}$n. When configured in Direct Bus (FM = LOW during a master reset), the $\overline{PR}$n bus provides packet ready status in 8 queue increments. The $\overline{PR}$n bus supports either Polled or Direct modes of operation. The $\overline{PR}$n mode of operation is configured through the Flag Mode (FM) bit during a Master Reset.

When the multi-queue is configured for packet mode operation, the device must also be configured for 36 bit write data bus and 36 bit read data bus. The two most significant bits of the 36-bit data bus are used as "packet markers". On the write port these are bits D34 (Transmit Start of Packet,) D35 (Transmit End of Packet) and on the read port Q34, Q35. All four bits are monitored by the packet control logic as data is written into and read out from the queues. The packet ready status for individual queues is then determined by the packet ready logic.

On the write port D34 is used to "mark" the first word being written into the selected queue as the "Transmit Start of Packet", TSOP. To further clarify, when the user requires a word being written to be marked as the start of a packet, the TSOP input (D34) must be HIGH for the same WCLK rising edge as the word that is written. The TSOP marker is stored in the queue along with the data it was written in until the word is read out of the queue via the read port.

On the write port D35 is used to "mark" the last word of the packet currently being written into the selected queue as the "Transmit End of Packet" TEOP. When the user requires a word being written to be marked as the end of a packet, the TEOP input must be HIGH for the same WCLK rising edge as the word that is written in. The TEOP marker is stored in the queue along with the data it was written in until the word is read out of the queue via the read port.

The packet ready logic monitors all start and end of packet markers both as they enter respective queues via the write port and as they exit queues via the

TABLE 5 — PACKET MODE VALID BYTE

D35/Q35	D34/Q34	D33/Q33	D32/Q32	D31/Q31	D23/Q23	D15/Q15	D7/Q7	D0/Q0
EOP	SOP	MOD 1	MOD 2	BYTE D	BYTE C	BYTE B	BYTE A	

TMOD1 (D33) RMOD1 (Q33)	TMOD2 (D32) RMOD2 (Q32)	VALID BYTES
0	0	A, B, C, D
0	1	A
1	0	A, B
1	1	A, B, C

5998 drw07

NOTE:
Packet Mode is only available when the Input Port and Output Port are 36 bits wide.

read port. The multi-queue internal logic increments and decrements a packet counter, which is provided for each queue. The functionality of the packet ready logic provides status as to whether at least one full packet of data is available within the selected queue. A partial packet in a queue is regarded as a packet not ready and $\overline{PR}$ (active LOW) will be HIGH. In Packet mode, no words can be read from a queue until at least one complete packet has been written into the queue, regardless of $\overline{REN}$. For example, if a TSOP has been written and some number of words later a TEOP is written a full packet of data is deemed to be available, and the $\overline{PR}$ flag and $\overline{OV}$ will go active LOW. Consequently if reads begin from a queue that has only one complete packet and the RSOP is detected on the output port as data is being read out, $\overline{PR}$ will go inactive HIGH. $\overline{OV}$ will remain LOW indicating there is still valid data being read out of that queue until the REOP is read. The user may proceed with the reading operation until the current packet has been read out and no further complete packets are available. If during that time another complete packet has been written into the queue and the $\overline{PR}$ flag will again gone active, then reads from the new packet may follow after the current packet has been completely read out.

The packet counters therefore look for start of packet markers followed by end of packet markers and regard data in between the TSOP and TEOP as a full packet of data. The packet monitoring has no limitation as to how many packets are written into a queue, the only constraint is the depth of the queue. Note, there is a minimum allowable packet size of four words, inclusive of the TSOP marker and TEOP marker.

The packet logic does expect a TSOP marker to be followed by a TEOP marker.

If a second TSOP marker is written after a first, it is ignored and the logic regards data between the first TSOP and the first subsequent TEOP as the full packet. The same is true for TEOP; a second consecutive TEOP mark is ignored. On the read side the user should regard a packet as being between the first RSOP and the first subsequent REOP and disregard consecutive RSOP markers and/or REOP markers. This is why a TEOP may be written twice, using the second TEOP as the "filler" word.

As an example, the user may also wish to implement the use of an "Almost End of Packet" (AEOP) marker. For example, the AEOP can be assigned to data input bit D33. The purpose of this AEOP marker is to provide an indicator that the end of packet is a fixed (known) number of reads away from the end of packet. This is a useful feature when due to latencies within the system, monitoring the REOP marker alone does not prevent "over reading" of the data from the queue selected. For example, an AEOP marker set 4 writes before the TEOP marker provides the device connected to the read port with and "almost end of packet" indication 4 cycles before the end of packet.

The AEOP can be set any number of words before the end of packet determined by user requirements or latencies involved in the system.

See Figure 18, *Reading in Packet Mode during a Queue Change*, Figure 19, *Data Input (Transmit) Packet Mode of Operation* and Figure 20, *Data Output (Receive) Packet Mode of Operation*.

PACKET MODE – MODULO OPERATION

The internal packet ready control logic performs no operation on these modulo bits, they are only informational bits that are passed through with the respective data byte(s).

When utilizing the multi-queue flow-control device in packet mode, the user may also want to consider the implementation of "Modulo" operation or "valid byte marking". Modulo operation may be useful when the packets being transferred through a queue are in a specific byte arrangement even though the data bus width is 36 bits. In Modulo operation the user can concatenate bytes to form a specific data string through the multi-queue device. A possible scenario is where a limited number of bytes are extracted from the packet for either analysis or filtered for security protection. This will only occur when the first 36 bit word of a packet is written in and the last 36 bit word of packet is written in. The modulo operation is a means by which the user can mark and identify specific data within the Queue.

On the write port data input bits, D32 (transmit modulo bit 2, TMOD2) and D33 (transmit modulo bit 1, TMOD1) can be used as data markers. An example of this could be to use D32 and D33 to code which bytes of a word are part of the packet that is also being marked as the "Start of Marker" or "End of Marker". Conversely on the read port when reading out these marked words, data outputs Q32 (receive modulo bit 2, RMOD2) and Q33 (receive modulo bit 1, RMOD1) will pass on the byte validity information for that word. Refer to Table 5 for one example of how the modulo bits may be setup and used. See Figure 19, *Data Input (Transmit) Packet Mode of Operation* and Figure 20, *Data Output (Receive) Packet Mode of Operation*.

NULL QUEUE OPERATION (OF THE READ PORT)

Pipelining of data to the output port enables the device to provide 100% bus utilization in standard mode. Data can be read out of the multi-queue flow-control device on every RCLK cycle regardless of queue switches or other operations. The device architecture is such that the pipeline is constantly filled with the next words in a selected queue to be read out, again providing 100% bus utilization. This type of architecture does assume that the user is constantly switching queues such that during a queue switch, the last data word required from the previous queue will fall through the pipeline to the output.

Note, that if reads cease at the empty boundary of a queue, then the last word will automatically flow through the pipeline to the output.

The Null Q operation is achieved by setting the Null Q signal HIGH during a queue select. Note that the read address bus RDADD[7:0] is a don't care. The Null Queue is a separate queue within the device and thus the maximum number of queues and memory is always available regardless of whether or not the Null queue is used. Also note that in expansion mode a user may want to use a dedicated null queue for each device. A null queue can be selected when no further reads are required from a previously selected queue. Changing to a null queue will continue to propagate data in the pipeline to the previous queue's output. The Null Q can remain selected until a data becomes available in another queue for reading. The Null-Q can be utilized in either standard or packet mode.

Note: If the user switches the read port to the null queue, this queue is seen as and treated as an empty queue, therefore after switching to the null queue the last word from the previous queue will remain in the output register and the $\overline{OV}$ flag will go HIGH, indicating data is not valid.

The Null queue operation only has significance to the read port of the multi-queue, it is a means to force data through the pipeline to the output. Null Q selection and operation has no meaning on the write port of the device. Also, refer to Figure 21, *Read Operation and Null Queue Select* for diagram.

$\overline{PAF}$n FLAG BUS OPERATION

The IDT72T51546/72T51556 multi-queue flow-control device can be configured for up to 32 queues, each queue having its own almost full status. An active queue has its flag status output to the discrete flags, $\overline{FF}$ and $\overline{PAF}$, on the write port. Queues that are not selected for a write operation can have their $\overline{PAF}$ status monitored via the $\overline{PAF}$n bus. The $\overline{PAF}$n flag bus is 8 bits wide, so that 8 queues at a time can have their status output to the bus. If 9 or more queues are setup within a device then there are 2 methods by which the device can share the bus between queues, "Direct" mode and "Polled" mode depending on the state of the FM (Flag Mode) input during a Master Reset. If 8 or less queues are setup within a device then each will have its own dedicated output from the bus. If 8 or less queues are setup in single device mode, it is recommended to configure the $\overline{PAF}$n bus to polled mode as it does not require using the write address (WRADD).

EXPANDING UP TO 256 QUEUES OR PROVIDING DEEPER QUEUES

Expansion can take place using either the standard mode or the packet mode. In the 32 queue multi-queue device, the WRADD address bus is 8 bits wide. The 5 Least Significant bits (LSbs) are used to address one of the 32 available queues within a single multi-queue device. The 3 Most Significant bits (MSbs) are used when a device is connected in expansion mode with up to 8 devices connected in width expansion, each device having its own 3-bit address. When logically expanded with multiple parts, each device is statically setup with a unique chip ID code on the ID pins, ID0, ID1, and ID2. A device is selected when the 3 Most Significant bits of the WRADD address bus matches a 3-bit ID code. The maximum logical expansion is 256 queues (32 queues x 8 devices) or a minimum of 8 queues (1 queue per device x 8 devices), each of the maximum size of the individual memory device.

Note: The WRADD bus is also used in conjunction with FSTR (almost full flag bus strobe), to address the almost full flag bus during direct mode of operation.

Refer to Table 1, for Write Address bus arrangement. Also, refer to Figure 12, *Full Flag Timing Expansion Mode*, Figure 14, *Output Valid Flag Timing (In Expansion Mode)*, and Figure 35, *Multi-Queue Expansion Diagram*, for timing diagrams.

BUS MATCHING OPERATION

Bus Matching operation between the input port and output port is available. During a master reset of the multi-queue the state of the three setup pins, BM (Bus Matching), IW (Input Width) and OW (Output Width) determine the input and output port bus widths as per the selections shown in Table 3, "Bus Matching Set-Up". 9 bit bytes, 18 bit words and 36 bit long words can be written into and read from the Queues provided that at least one of the ports is setup for x36 operation. When writing to or reading from the multi-queue in a bus matching mode, the device orders data in a "Little Endian" format. See Figure 4, *Bus Matching Byte Arrangement* for details.

The Full flag and Almost Full flag operation is always based on writes and reads of data widths determined by the write port width. For example, if the input port is x36 and the output port is x9, then four data reads from a full queue will be required to cause the full flag to go HIGH (queue not full). Conversely, the Output Valid flag and Almost Empty flag operations are always based on writes and reads of data widths determined by the read port. For example, if the input port is x18 and the output port is x36, two write operations will be required to cause the output valid flag of an empty queue to go LOW, output valid (queue is not empty).

Note, that the input port serves all queues within a device, as does the output port, therefore the input bus width to all queues is equal (determined by the input port size) and the output bus width from all queues is equal (determined by the output port size).

TABLE 3 — BUS-MATCHING SET-UP

x36 DEVICE

BM	IW	OW	Write Port	Read Port
0	X	X	x36	x36
1	0	0	x36	x18
1	0	1	x36	x9
1	1	0	x18	x36
1	1	1	x9	x36

FULL FLAG OPERATION

The multi-queue flow-control device provides a single Full Flag output, $\overline{FF}$. The $\overline{FF}$ flag output provides a full status of the queue currently selected on the write port for write operations. Internally the multi-queue flow-control device monitors and maintains a status of the full condition of all queues within it, however only the queue that is selected for write operations has its full status output to the $\overline{FF}$ flag. This dedicated flag is often referred to as the "active queue full flag".

When queue switches are being made on the write port, the $\overline{FF}$ flag output will switch to the new queue and provide the user with the new queue status, on the cycle after a new queue selection is made. The user then has a full status for the new queue one cycle ahead of the WCLK rising edge that data can be written into the new queue. That is, a new queue can be selected on the write port via the WRADD bus, WADEN enable and a rising edge of WCLK. On the second rising edge of WCLK, the $\overline{FF}$ flag output will show the full status of the newly selected queue. On the third rising edge of WCLK following the queue selection, data can be written into the newly selected queue provided that data and enable setup & hold times are met.

Note, the $\overline{FF}$ flag will provide status of a newly selected queue two WCLK cycle after queue selection, which is one cycle before data can be written to that queue. This prevents the user from writing data to a queue that is full, (assuming that a queue switch has been made to a queue that is actually full).

The $\overline{FF}$ flag is synchronous to the WCLK and all transitions of the $\overline{FF}$ flag occur based on a rising edge of WCLK. Internally the multi-queue device monitors and keeps a record of the full status for all queues. It is possible that the status of a $\overline{FF}$ flag maybe changing internally even though that flag is not the active queue flag (selected on the write port). A queue selected on the read port may experience a change of its internal full flag status based on read operations.

See Figure 10, *Write Queue Select, Write Operation and Full Flag Operation* and Figure 12, *Full Flag Timing in Expansion Mode* for timing information.

EXPANSION MODE - FULL FLAG OPERATION

When multi-queue devices are connected in Expansion mode the $\overline{FF}$ flags of all devices should be connected together, such that a system controller monitoring and managing the multi-queue devices write port only looks at a single $\overline{FF}$ flag (as opposed to a discrete $\overline{FF}$ flag for each device). This $\overline{FF}$ flag is only pertinent to the queue being selected for write operations at that time. Remember, that when in expansion mode only one multi-queue device can be written to at any moment in time, thus the $\overline{FF}$ flag provides status of the active queue on the write port.

This connection of flag outputs to create a single flag requires that the $\overline{FF}$ flag output have a High-Impedance capability, such that when a queue selection is made only a single device drives the $\overline{FF}$ flag bus and all other $\overline{FF}$ flag outputs connected to the $\overline{FF}$ flag bus are placed into High-Impedance. The user does not have to select this High-Impedance state, a given multi-queue flow-control device will automatically place its $\overline{FF}$ flag output into High-Impedance when none of its queues are selected for write operations.

When queues within a single device are selected for write operations, the $\overline{FF}$ flag output of that device will maintain control of the $\overline{FF}$ flag bus. Its $\overline{FF}$ flag will simply update between queue switches to show the respective queue full status.

The multi-queue device places its $\overline{FF}$ flag output into High-Impedance based on the 3 bit ID code found in the 3 most significant bits of the write queue address bus, WRADD. If the 3 most significant bits of WRADD match the 3 bit ID code setup on the static inputs, ID0, ID1 and ID2 then the $\overline{FF}$ flag output of the respective device will be in a Low-Impedance state. If they do not match, then the $\overline{FF}$ flag output of the respective device will be in a High-Impedance state. See Figure 12, *Full Flag Timing in Expansion Mode* for details of flag operation, including when more than one device is connected in expansion.

OUTPUT VALID FLAG OPERATION

The multi-queue flow-control device provides a single Output Valid flag output, $\overline{OV}$. The $\overline{OV}$ provides an empty status or data output valid status for the data word currently available on the output register of the read port. The rising edge of an RCLK cycle that places new data onto the output register of the read port, also updates the $\overline{OV}$ flag to show whether or not that new data word is actually valid. Internally the multi-queue flow-control device monitors and maintains a status of the empty condition of all queues within it, however only the queue that is selected for read operations has its output valid (empty) status output to the $\overline{OV}$ flag, giving a valid status for the word being read at that time.

The nature of the first word fall through operation means that when the last data word is read from a selected queue, the $\overline{OV}$ flag will go HIGH on the next enabled read, that is, on the next rising edge of RCLK while $\overline{REN}$ is LOW.

When queue switches are being made on the read port, the $\overline{OV}$ flag will switch to show status of the new queue in line with the data output from the new queue. When a queue selection is made the first data from that queue will appear on the Qout data outputs 3 RCLK cycles later, the $\overline{OV}$ will change state to indicate validity of the data from the newly selected queue on this 3rd RCLK cycle also. The previous cycles will continue to output data from the previous queue and the $\overline{OV}$ flag will indicate the status of those outputs. Again, the $\overline{OV}$ flag always indicates status for the data currently present on the output register.

The $\overline{OV}$ flag is synchronous to the RCLK and all transitions of the $\overline{OV}$ flag occur based on a rising edge of RCLK. Internally the multi-queue device monitors and keeps a record of the output valid (empty) status for all queues. It is possible that the status of an $\overline{OV}$ flag may be changing internally even though that respective flag is not the active queue flag (selected on the read port). A queue selected on the write port may experience a change of its internal $\overline{OV}$ flag status based on write operations, that is, data may be written into that queue causing it to become "not empty".

See Figure 13, *Read Queue Select, Read Operation* and Figure 14, *Output Valid Flag Timing* for details of the timing.

EXPANSION MODE – OUTPUT VALID FLAG OPERATION

When multi-queue devices are connected in Expansion mode, the $\overline{OV}$ flags of all devices should be connected together, such that a system controller monitoring and managing the multi-queue devices read port only looks at a single $\overline{OV}$ flag (as opposed to a discrete $\overline{OV}$ flag for each device). This $\overline{OV}$ flag is only pertinent to the queue being selected for read operations at that time. Remember, that when in expansion mode only one multi-queue device can be read from at any moment in time, thus the $\overline{OV}$ flag provides status of the active queue on the read port.

This connection of flag outputs to create a single flag requires that the $\overline{OV}$ flag output have a High-Impedance capability, such that when a queue selection is made only a single device drives the $\overline{OV}$ flag bus and all other $\overline{OV}$ flag outputs connected to the $\overline{OV}$ flag bus are placed into High-Impedance. The user does not have to select this High-Impedance state, a given multi-queue flow-control device will automatically place its $\overline{OV}$ flag output into High-Impedance when none of its queues are selected for read operations.

When queues within a single device are selected for read operations, the $\overline{OV}$ flag output of that device will maintain control of the $\overline{OV}$ flag bus. Its $\overline{OV}$ flag will simply update between queue switches to show the respective queue output valid status.

The multi-queue device places its $\overline{OV}$ flag output into High-Impedance based on the 3 bit ID code found in the 3 most significant bits of the read queue address bus, RDADD. If the 3 most significant bits of RDADD match the 3 bit ID code setup on the static inputs, ID0, ID1 and ID2 then the $\overline{OV}$ flag output of the respective device will be in a Low-Impedance state. If they do not match, then the $\overline{OV}$ flag output of the respective device will be in a High-Impedance state. See Figure 14, *Output Valid Flag Timing* for details of flag operation, including when more than one device is connected in expansion.

ALMOST FULL FLAG

As previously mentioned the multi-queue flow-control device provides a single Programmable Almost Full flag output, $\overline{PAF}$. The $\overline{PAF}$ flag output provides a status of the almost full condition for the active queue currently selected on the write port for write operations. Internally the multi-queue flow-control device monitors and maintains a status of the almost full condition of all queues within it, however only the queue that is selected for write operations has its full status output to the $\overline{PAF}$ flag. This dedicated flag is often referred to as the "active queue almost full flag". The position of the $\overline{PAF}$ flag boundary within a queue can be at any point within that queues depth. This location can be user programmed via the serial port or one of the default values (8 or 128) can be selected if the user has performed default programming.

As mentioned, every queue within a multi-queue device has its own almost full status, when a queue is selected on the write port, this status is output via the $\overline{PAF}$ flag. The $\overline{PAF}$ flag value for each queue is programmed during multi-queue device programming (along with the number of queues, queue depths and almost empty values). The $\overline{PAF}$ offset value, m, for a respective queue can be programmed to be anywhere between '0' and 'D', where 'D' is the total memory depth for that queue. The $\overline{PAF}$ value of different queues within the same device can be different values.

When queue switches are being made on the write port, the $\overline{PAF}$ flag output will switch to the new queue and provide the user with the new queue status, on the third cycle after a new queue selection is made, on the same WCLK cycle that data can actually be written to the new queue. That is, a new queue can be selected on the write port via the WRADD bus, WADEN enable and a rising edge of WCLK. On the third rising edge of WCLK following a queue selection, the $\overline{PAF}$ flag output will show the full status of the newly selected queue. The $\overline{PAF}$ is flag output is triple register buffered, so when a write operation occurs at the almost full boundary causing the selected queue status to go almost full the $\overline{PAF}$ will go LOW 3 WCLK cycles after the write. The same is true when a read occurs, there will be a 3 WCLK cycle delay after the read operation.

So the $\overline{PAF}$ flag delays are:

from a write operation to $\overline{PAF}$ flag LOW is 2 WCLK + t_{WAF}

The delay from a read operation to $\overline{PAF}$ flag HIGH is t_{SKEW2} + WCLK + t_{WAF}

Note, if t_{SKEW} is violated there will be one added WCLK cycle delay.

The $\overline{PAF}$ flag is synchronous to the WCLK and all transitions of the $\overline{PAF}$ flag occur based on a rising edge of WCLK. Internally the multi-queue device monitors and keeps a record of the almost full status for all queues. It is possible that the status of a $\overline{PAF}$ flag maybe changing internally even though that flag is not the active queue flag (selected on the write port). A queue selected on the read port may experience a change of its internal almost full flag status based on read operations. The multi-queue flow-control device also provides a duplicate of the $\overline{PAF}$ flag on the $\overline{PAF}$[7:0] flag bus, this will be discussed in detail in a later section of the data sheet.

See Figures 23 and 24 for Almost Full flag timing and queue switching.

ALMOST EMPTY FLAG

As previously mentioned the multi-queue flow-control device provides a single Programmable Almost Empty flag output, $\overline{PAE}$. The $\overline{PAE}$ flag output provides a status of the almost empty condition for the active queue currently selected on the read port for read operations. Internally the multi-queue flow-control device monitors and maintains a status of the almost empty condition of all queues within it, however only the queue that is selected for read operations has its empty status output to the $\overline{PAE}$ flag. This dedicated flag is often referred to as the "active queue almost empty flag". The position of the $\overline{PAE}$ flag boundary within a queue can be at any point within that queues depth. This location can be user programmed via the serial port or one of the default values (8 or 128) can be selected if the user has performed default programming.

As mentioned, every queue within a multi-queue device has its own almost empty status, when a queue is selected on the read port, this status is output via the $\overline{PAE}$ flag. The $\overline{PAE}$ flag value for each queue is programmed during multi-queue device programming (along with the number of queues, queue depths and almost full values). The $\overline{PAE}$ offset value, n, for a respective queue can be programmed to be anywhere between '0' and 'D', where 'D' is the total memory depth for that queue. The $\overline{PAE}$ value of different queues within the same device can be different values.

When queue switches are being made on the read port, the $\overline{PAE}$ flag output will switch to the new queue and provide the user with the new queue status, on the third cycle after a new queue selection is made, on the same RCLK cycle that data actually falls through to the output register from the new queue. That is, a new queue can be selected on the read port via the RDADD bus, RADEN enable and a rising edge of RCLK. On the third rising edge of RCLK following a queue selection, the data word from the new queue will be available at the output register and the $\overline{PAE}$ flag output will show the empty status of the newly selected queue. The $\overline{PAE}$ is flag output is triple register buffered, so when a read operation occurs at the almost empty boundary causing the selected queue status to go almost empty the $\overline{PAE}$ will go LOW 3 RCLK cycles after the read. The same is true when a write occurs, there will be a 3 RCLK cycle delay after the write operation.

So the $\overline{PAE}$ flag delays are:

from a read operation to $\overline{PAE}$ flag LOW is 2 RCLK + t_{RAE}

The delay from a write operation to $\overline{PAE}$ flag HIGH is t_{SKEW2} + RCLK + t_{RAE}

Note, if t_{SKEW} is violated there will be one added RCLK cycle delay.

The $\overline{PAE}$ flag is synchronous to the RCLK and all transitions of the $\overline{PAE}$ flag occur based on a rising edge of RCLK. Internally the multi-queue device monitors and keeps a record of the almost empty status for all queues. It is possible that the status of a $\overline{PAE}$ flag maybe changing internally even though that flag is not the active queue flag (selected on the read port). A queue selected on the write port may experience a change of its internal almost empty flag status based on write operations. The multi-queue flow-control device also provides a duplicate of the $\overline{PAE}$ flag on the $\overline{PAE}$[7:0] flag bus, this will be discussed in detail in a later section of the data sheet.

See Figures 25 and 26 for Almost Empty flag timing and queue switching.

POWER DOWN (PD)

This device has a power down feature intended for reducing power consumption for HSTL/eHSTL configured inputs when the device is idle for a long period of time. By entering the power down state certain inputs can be disabled, thereby significantly reducing the power consumption of the part. All $\overline{WEN}$ and $\overline{REN}$ signals must be disabled for a minimum of four WCLK and RCLK cycles before activating the power down signal. The power down signal is asynchronous and needs to be held LOW throughout the desired power down time. During power down, the following conditions for the inputs/outputs signals are:

- All data in Queue(s) memory are retained.
- All data inputs become inactive.
- All write and read pointers maintain their last value before power down.
- All enables, chip selects, and clock input pins become inactive.
- All data outputs become inactive and enter high-impedance state.
- All flag outputs will maintain their current states before power down.
- All programmable flag offsets maintain their values.
- All echo clocks and enables will become inactive and enter high-impedance state.
- The serial programming and JTAG port will become inactive and enter high-impedance state.
- All setup and configuration CMOS static inputs are not affected, as these pins are tied to a known value and do not toggle during operation.

All internal counters, registers, and flags will remain unchanged and maintain their current state prior to power down. Clock inputs can be continuous and free-running during power down, but will have no affect on the part. However, it is recommended that the clock inputs be low when the power down is active. To exit power down state and resume normal operations, disable the power down signal by bringing it HIGH. There must be a minimum of 1µs waiting period before read and write operations can resume. The device will continue from where it had stopped and no form of reset is required after exiting power down state. The power down feature does not provide any power savings when the inputs are configured for LVTTL operation. However, it will reduce the current for I/Os that are not tied directly to V_{CC} or GND. See Figure 34, *Power Down Operation,* for the associated timing diagram.

TABLE 4 — FLAG OPERATION BOUNDARIES & TIMING

Output Valid, $\overline{OV}$ Flag Boundary	
I/O Set-Up	**$\overline{OV}$ Boundary Condition**
In36 to out36 (Almost Empty Mode) (Both ports selected for same queue when the 1st Word is written in)	$\overline{OV}$ Goes LOW after 1st Write (see note 1 below for timing)
In36 to out36 (Packet Mode) (Both ports selected for same queue when the 1st Word is written in)	$\overline{OV}$ Goes LOW after 1st Write (see note 2 below for timing)
In36 to out18 (Both ports selected for same queue when the 1st Word is written in)	$\overline{OV}$ Goes LOW after 1st Write (see note 1 below for timing)
In36 to out9 (Both ports selected for same queue when the 1st Word is written in)	$\overline{OV}$ Goes LOW after 1st Write (see note 1 below for timing)
In18 to out36 (Both ports selected for same queue when the 1st Word is written in)	$\overline{OV}$ Goes LOW after 1st Write (see note 1 below for timing)
In9 to out36 (Both ports selected for same queue when the 1st Word is written in)	$\overline{OV}$ Goes LOW after 1st Write (see note 1 below for timing)

NOTE:
1. **$\overline{OV}$ Timing**
Assertion:
Write to $\overline{OV}$ LOW: t_{SKEW1} + RCLK + t_{ROV}
If t_{SKEW1} is violated there may be 1 added clock: t_{SKEW1} + 2 RCLK + t_{ROV}
De-assertion:
Read Operation to $\overline{OV}$ HIGH: t_{ROV}

2. **$\overline{OV}$ Timing when in Packet Mode (36 in to 36 out only)**
Assertion:
Write to $\overline{OV}$ LOW: t_{SKEW4} + RCLK + t_{ROV}
If t_{SKEW4} is violated there may be 1 added clock: t_{SKEW4} + 2 RCLK + t_{ROV}
De-assertion:
Read Operation to $\overline{OV}$ HIGH: t_{ROV}

Full Flag, $\overline{FF}$ Boundary	
I/O Set-Up	**$\overline{FF}$ Boundary Condition**
In36 to out36 (Both ports selected for same queue when the 1st Word is written in)	$\overline{FF}$ Goes LOW after D+1 Writes (see note below for timing)
In36 to out36 (Write port only selected for queue when the 1st Word is written in)	$\overline{FF}$ Goes LOW after D Writes (see note below for timing)
In36 to out18 (Both ports selected for same queue when the 1st Word is written in)	$\overline{FF}$ Goes LOW after D Writes (see note below for timing)
In36 to out18 (Write port only selected for queue when the 1st Word is written in)	$\overline{FF}$ Goes LOW after D Writes (see note below for timing)
In36 to out9 (Both ports selected for same queue when the 1st Word is written in)	$\overline{FF}$ Goes LOW after D Writes (see note below for timing)
In36 to out9 (Write port only selected for queue when the 1st Word is written in)	$\overline{FF}$ Goes LOW after D Writes (see note below for timing)
In18 to out36 (Both ports selected for same queue when the 1st Word is written in)	$\overline{FF}$ Goes LOW after ([D+1] x 2) Writes (see note below for timing)
In18 to out36 (Write port only selected for queue when the 1st Word is written in)	$\overline{FF}$ Goes LOW after (D x 2) Writes (see note below for timing)
In9 to out36 (Both ports selected for same queue when the 1st Word is written in)	$\overline{FF}$ Goes LOW after ([D+1] x 4) Writes (see note below for timing)
In9 to out36 (Write port only selected for queue when the 1st Word is written in)	$\overline{FF}$ Goes LOW after (D x 4) Writes (see note below for timing)

NOTE:
D = Queue Depth
$\overline{FF}$ Timing
Assertion:
Write Operation to $\overline{FF}$ LOW: t_{WFF}
De-assertion:
Read to $\overline{FF}$ HIGH: t_{SKEW1} + t_{WFF}
If t_{SKEW1} is violated there may be 1 added clock: t_{SKEW1}+WCLK +t_{WFF}

Programmable Almost Full Flag, $\overline{PAF}$ & $\overline{PAF}$n Bus Boundary	
I/O Set-Up	**$\overline{PAF}$ & $\overline{PAF}$n Boundary**
in36 to out36 (Both ports selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAF}$/$\overline{PAF}$n Goes LOW after D+1-m Writes (see note below for timing)
in36 to out36 (Write port only selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAF}$/$\overline{PAF}$n Goes LOW after D-m Writes (see note below for timing)
in36 to out18	$\overline{PAF}$/$\overline{PAF}$n Goes LOW after D-m Writes (see below for timing)
in36 to out9	$\overline{PAF}$/$\overline{PAF}$n Goes LOW after D-m Writes (see below for timing)
in18 to out36	$\overline{PAF}$/$\overline{PAF}$n Goes LOW after ([D+1-m] x 2) Writes (see note below for timing)
in9 to out36	$\overline{PAF}$/$\overline{PAF}$n Goes LOW after ([D+1-m] x 4) Writes (see note below for timing)

NOTE:
D = Queue Depth
m = Almost Full Offset value.
Default values: if DF is LOW at Master Reset then m = 8
if DF is HIGH at Master Reset then m= 128
$\overline{PAF}$ Timing
Assertion: Write Operation to $\overline{PAF}$ LOW: 2 WCLK + t_{WAF}
De-assertion: Read to $\overline{PAF}$ HIGH: t_{SKEW2} + WCLK + t_{WAF}
If t_{SKEW2} is violated there may be 1 added clock: t_{SKEW2} + 2 WCLK + t_{WAF}
$\overline{PAF}$n Timing
Assertion: Write Operation to $\overline{PAF}$n LOW: 2 WCLK* + t_{PAF}
De-assertion: Read to $\overline{PAF}$n HIGH: t_{SKEW3} + WCLK* + t_{PAF}
If t_{SKEW3} is violated there may be 1 added clock: t_{SKEW3} + 2 WCLK* + t_{PAF}
* If a queue switch is occurring on the write port at the point of flag assertion or de-assertion there may be one additional WCLK clock cycle delay.

TABLE 4 — FLAG OPERATION BOUNDARIES & TIMING (CONTINUED)

Programmable Almost Empty Flag, $\overline{PAE}$ Boundary	
I/O Set-Up	**$\overline{PAE}$ Assertion**
in36 to out36 (Both ports selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$ Goes HIGH after n+2 Writes (see note below for timing)
in36 to out18 (Both ports selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$ Goes HIGH after n+1 Writes (see note below for timing)
in36 to out9 (Both ports selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$ Goes HIGH after n+1 Writes (see note below for timing)
in18 to out36 (Both ports selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$ Goes HIGH after ([n+2] x 2) Writes (see note below for timing)
in9 to out36 (Both ports selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$ Goes HIGH after ([n+2] x 4) Writes (see note below for timing)

NOTE:
n = Almost Empty Offset value.
Default values: if DF is LOW at Master Reset then n = 8
if DF is HIGH at Master Reset then n = 128

$\overline{PAE}$ Timing
Assertion: Read Operation to $\overline{PAE}$ LOW: 2 RCLK + t_{RAE}
De-assertion: Write to $\overline{PAE}$ HIGH: t_{SKEW2} + RCLK + t_{RAE}
If t_{SKEW2} is violated there may be 1 added clock: t_{SKEW2} + 2 RCLK + t_{RAE}

Programmable Almost Empty Flag Bus, $\overline{PAE}$n Boundary	
I/O Set-Up	**$\overline{PAE}$n Boundary Condition**
in36 to out36 (Both ports selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$n Goes HIGH after n+2 Writes (see note below for timing)
in36 to out36 (Write port only selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$n Goes HIGH after n+1 Writes (see note below for timing)
in36 to out18	$\overline{PAE}$n Goes HIGH after n+1 Writes (see below for timing)
in36 to out9	$\overline{PAE}$n Goes HIGH after n+1 Writes (see below for timing)
in18 to out36 (Both ports selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$n Goes HIGH after ([n+2] x 2) Writes (see note below for timing)
in18 to out36 (Write port only selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$n Goes HIGH after ([n+1] x 2) Writes (see note below for timing)
in9 to out36 (Both ports selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$n Goes HIGH after ([n+2] x 4) Writes (see note below for timing)
in9 to out36 (Write port only selected for same queue when the 1st Word is written in until the boundary is reached)	$\overline{PAE}$n Goes HIGH after ([n+1] x 4) Writes (see note below for timing)

NOTE:
n = Almost Empty Offset value.
Default values: if DF is LOW at Master Reset then n = 8
if DF is HIGH at Master Reset then n = 128

$\overline{PAE}$n Timing
Assertion: Read Operation to $\overline{PAE}$n LOW: 2 RCLK* + t_{PAE}
De-assertion: Write to $\overline{PAE}$n HIGH: t_{SKEW3} + RCLK* + t_{PAE}
If t_{SKEW3} is violated there may be 1 added clock: t_{SKEW3} + 2 RCLK* + t_{PAE}

* If a queue switch is occurring on the read port at the point of flag assertion or de-assertion there may be one additional RCLK clock cycle delay.

PACKET READY FLAG, $\overline{PR}$ BOUNDARY

Assertion:
Both the rising and falling edges of $\overline{PR}$ are synchronous to RCLK.
$\overline{PR}$ Falling Edge occurs upon writing the first TEOP marker, on input D35, (assuming a TSOP marker, on input D34 has previously been written). i.e. a complete packet is available within a queue.

Timing:
From WCLK rising edge writing the TEOP word $\overline{PR}$ goes LOW after: t_{SKEW4} + 2 RCLK + t_{PR}
If t_{SKEW4} is violated:
$\overline{PR}$ goes LOW after t_{SKEW4} + 3 RCLK + t_{PR}
(Please refer to Figure 19, *Data Input (Transmit) Packet Mode of Operation,* for timing diagram).

De-assertion:
$\overline{PR}$ Rising Edge occurs upon reading the last RSOP marker, from output Q34. i.e. there are no more complete packets available within the queue.

Timing:
From RCLK rising edge Reading the RSOP word the $\overline{PR}$ goes HIGH after: 3 RCLK + t_{PR}
(Please refer to Figure 20, *Data Output (Receive) Packet Mode of Operation* for timing diagram).

PACKET READY FLAG BUS, $\overline{PR}$n BOUNDARY

Assertion:
Both the rising and falling edges of $\overline{PR}$n are synchronous to RCLK.
$\overline{PR}$n Falling Edge occurs upon writing the first TEOP marker, on input D35, (assuming a TSOP marker, on input D34 has previously been written). i.e. a complete packet is available within a queue.

Timing:
From WCLK rising edge writing the TEOP word $\overline{PR}$ goes LOW after: t_{SKEW4} + 2 RCLK* + t_{PAE}
If t_{SKEW4} is violated $\overline{PR}$n goes LOW after t_{SKEW4} + 3 RCLK* + t_{PAE}
*If a queue switch is occurring on the read port at the point of flag assertion there may be one additional RCLK clock cycle delay.

De-assertion:
$\overline{PR}$ Rising Edge occurs upon reading the last RSOP marker, from output Q34. i.e. there are no more complete packets available within the queue.

Timing:
From RCLK rising edge Reading the RSOP word the $\overline{PR}$ goes HIGH after: 3 RCLK* + t_{PAE}
*If a queue switch is occurring on the read port at the point of flag assertion or de-assertion there may be one additional RCLK clock cycle delay.

$\overline{PAF}$n - DIRECT BUS

If FM is LOW at master reset then the $\overline{PAF}$n bus operates in Direct (addressed) mode. In direct mode the user can address the quadrant of queues they require to be placed on to the $\overline{PAF}$n bus. For example, consider the operation of the $\overline{PAF}$n bus when 26 queues have been setup. To output status of the first quadrant, Queue[0:7] the WRADD bus is used in conjunction with the FSTR ($\overline{PAF}$ flag strobe) input and WCLK. The address present on the 2 least significant bits of the WRADD bus with FSTR HIGH will be selected as the quadrant address on a rising edge of WCLK. So to address quadrant 1, Queue[0:7] the WRADD bus should be loaded with "xxxxxx00", the $\overline{PAF}$n bus will change status to show the new quadrant selected 1 WCLK cycle after quadrant selection. $\overline{PAF}$n[0:7] gets status of queues, Queue[0:7] respectively.

To address the second quadrant, Queue[8:15], the WRADD address is "xxxxxx01". $\overline{PAF}$n[0:7] gets status of queues, Queue[8:15] respectively. To address the third quadrant, Queue[16:23], the WRADD address is "xxxxxx10". $\overline{PAF}$[0:7] gets status of queues, Queue[16:23] respectively. To address the fourth quadrant, Queue[24:31], the WRADD address is "xxxxxx11". $\overline{PAF}$[0:1] gets status of queues, Queue[24:25] respectively. Remember, only 26 queues were setup, so when quadrant 4 is selected the unused outputs $\overline{PAF}$[2:7] will be don't care states.

Note, that if a read or write operation is occurring to a specific queue, say queue 'x' on the same cycle as a quadrant switch which will include the queue 'x', then there may be an extra WCLK cycle delay before that queues status is correctly shown on the respective output of the $\overline{PAF}$n bus. However, the active $\overline{PAF}$ flag will show correct status at all times.

Quadrants can be selected on consecutive clock cycles, that is the quadrant on the $\overline{PAF}$n bus can change every WCLK cycle. Also, data present on the input bus, Din, can be written into a Queue on the same WCLK rising edge that a quadrant is being selected, the only restriction being that a write queue selection and $\overline{PAF}$n quadrant selection cannot be made on the same cycle.

If 8 or less queues are setup then queues, Queue[0:7] have their $\overline{PAF}$ status output on $\overline{PAF}$[0:7] constantly.

When the multi-queue devices are connected in expansion of more than one device the $\overline{PAF}$n busses of all devices are connected together, when switching between quadrants of different devices the user must utilize the 3 most significant bits of the WRADD address bus (as well as the 2 LSB's). These 3 MSB's correspond to the device ID inputs, which are the static inputs, ID0, ID1 & ID2.

Please refer to Figure 29 *$\overline{PAF}$n - Direct Mode Quadrant Selection* for timing information. Also refer to Table 1, *Write Address Bus, WRADD*.

$\overline{PAF}$n – POLLED BUS

If FM is HIGH at master reset then the $\overline{PAF}$n bus operates in Polled (looped) mode. In polled mode the $\overline{PAF}$n bus automatically cycles through the 4 quadrants within the device regardless of how many queues have been setup in the part. Every rising edge of the WCLK causes the next quadrant to be loaded on the $\overline{PAF}$n bus. The device configured as the master (MAST input tied HIGH), will take control of the $\overline{PAF}$n after $\overline{MRS}$ goes LOW. For the whole WCLK cycle that the first quadrant is on $\overline{PAF}$n the FSYNC ($\overline{PAF}$n bus sync) output will be HIGH, for all other quadrants, this FSYNC output will be LOW. This FSYNC output provides the user with a mark with which they can synchronize to the $\overline{PAF}$n bus, FSYNC is always HIGH for the WCLK cycle that the first quadrant of a device is present on the $\overline{PAF}$n bus.

When devices are connected in expansion mode, only one device will be set as the Master, MAST input tied HIGH, all other devices will have MAST tied LOW. The master device is the first device to take control of the $\overline{PAF}$n bus and will place its first quadrant on the bus on the rising edge of WCLK after the $\overline{MRS}$ input goes HIGH. For the next 3 WCLK cycles the master device will maintain control of the $\overline{PAF}$n bus and cycle its quadrants through it, all other devices hold their $\overline{PAF}$n outputs in High-Impedance. When the master device has cycled all of its quadrants it passes a token to the next device in the chain and that device assumes control of the $\overline{PAF}$n bus and then cycles its quadrants and so on, the $\overline{PAF}$n bus control token being passed on from device to device. This token passing is done via the FXO outputs and FXI inputs of the devices ("$\overline{PAF}$ Expansion Out" and "$\overline{PAF}$ Expansion In"). The FXO output of the master device connects to the FXI of the second device in the chain and the FXO of the second connects to the FXI of the third and so on. The final device in a chain has its FXO connected to the FXI of the first device, so that once the $\overline{PAF}$n bus has cycled through all quadrants of all devices, control of the $\overline{PAF}$n will pass to the master device again and so on. The FSYNC of each respective device will operate independently and simply indicate when that respective device has taken control of the bus and is placing its first quadrant on to the $\overline{PAF}$n bus.

When operating in single device mode the FXI input must be connected to the FXO output of the same device. In single device mode a token is still required to be passed into the device for accessing the $\overline{PAF}$n bus.

Please refer to Figure 32, *$\overline{PAF}$n Bus – Polled Mode* for timing information.

$\overline{PAE}$n/$\overline{PR}$n FLAG BUS OPERATION

The IDT72T51546/72T51556 multi-queue flow-control device can be configured for up to 32 queues, each queue having its own almost empty/packet ready status. An active queue has its flag status output to the discrete flags, $\overline{OV}$, $\overline{PAE}$ and $\overline{PR}$, on the read port. Queues that are not selected for a read operation can have their $\overline{PAE}$/$\overline{PR}$ status monitored via the $\overline{PAE}$n/$\overline{PR}$n bus. The $\overline{PAE}$n/$\overline{PR}$n flag bus is 8 bits wide, so that 8 queues at a time can have their status output to the bus. If 9 or more queues are setup within a device then there are 2 methods by which the device can share the bus between queues, "Direct" mode and "Polled" mode depending on the state of the FM (Flag Mode) input during a Master Reset. If 8 or less queues are setup within a device then each will have its own dedicated output from the bus. If 8 or less queues are setup in single device mode, it is recommended to configure the $\overline{PAF}$n bus to polled mode as it does not require using the write address (WRADD).

$\overline{PAE}$n/$\overline{PR}$n - DIRECT BUS

If FM is LOW at master reset then the $\overline{PAE}$n/$\overline{PR}$n bus operates in Direct (addressed) mode. In direct mode the user can address the quadrant of queues they require to be placed on to the $\overline{PAE}$n/$\overline{PR}$n bus. For example, consider the operation of the $\overline{PAE}$n/$\overline{PR}$n bus when 26 queues have been setup. To output status of the first quadrant, Queue[0:7] the RDADD bus is used in conjunction with the ESTR ($\overline{PAE}$/$\overline{PR}$ flag strobe) input and RCLK. The address present on the 2 least significant bits of the RDADD bus with ESTR HIGH will be selected as the quadrant address on a rising edge of RCLK. So to address quadrant 1, Queue[0:7] the RDADD bus should be loaded with "xxxxxx00", the $\overline{PAE}$n/$\overline{PR}$n bus will change status to show the new quadrant selected 1 RCLK cycle after quadrant selection. $\overline{PAE}$n[0:7] gets status of queues, Queue[0:7] respectively.

To address the second quadrant, Queue[8:15], the RDADD address is "xxxxxx01". $\overline{PAE}$n[0:7] gets status of queues, Queue[8:15] respectively. To address the third quadrant, Queue[16:23], the RDADD address is "xxxxxx10". $\overline{PAE}$[0:7] gets status of queues, Queue[16:23] respectively. To address the fourth quadrant, Queue[24:31], the RDADD address is "xxxxxx11". $\overline{PAE}$[0:1] gets status of queues, Queue[24:25] respectively. Remember, only 26 queues were setup, so when quadrant 4 is selected the unused outputs $\overline{PAE}$[2:7] will be don't care states.

Note, that if a read or write operation is occurring to a specific queue, say queue 'x' on the same cycle as a quadrant switch which will include the queue 'x', then there may be an extra RCLK cycle delay before that queues status is correctly shown on the respective output of the $\overline{PAE}$n/$\overline{PR}$n bus.

Quadrants can be selected on consecutive clock cycles, that is the quadrant on the $\overline{PAE}$n/$\overline{PR}$n bus can change every RCLK cycle. Also, data can be read out of a Queue on the same RCLK rising edge that a quadrant is being selected, the only restriction being that a read queue selection and $\overline{PAE}$n/$\overline{PR}$n quadrant selection cannot be made on the same RCLK cycle.

If 8 or less queues are setup then queues, Queue[0:7] have their $\overline{PAE}$/$\overline{PR}$ status output on $\overline{PAE}$[0:7] constantly.

When the multi-queue devices are connected in expansion of more than one device the $\overline{PAE}$n/$\overline{PR}$n busses of all devices are connected together, when switching between quadrants of different devices the user must utilize the 3 most significant bits of the RDADD address bus (as well as the 2 LSB's). These 3 MSB's correspond to the device ID inputs, which are the static inputs, ID0, ID1 & ID2.

Please refer to Figure 28, *$\overline{PAE}$n/$\overline{PR}$n - Direct Mode Quadrant Selection* for timing information. Also refer to Table 2, *Read Address Bus, RDADD*.

$\overline{PAE}$n – POLLED BUS

If FM is HIGH at master reset then the $\overline{PAE}$n/$\overline{PR}$n bus operates in Polled (looped) mode. In polled mode the $\overline{PAE}$n/$\overline{PR}$n bus automatically cycles through the 4 quadrants within the device regardless of how many queues have been setup in the part. Every rising edge of the RCLK causes the next quadrant to be loaded on the $\overline{PAE}$n/$\overline{PR}$n bus. The device configured as the master (MAST input tied HIGH), will take control of the $\overline{PAE}$n/$\overline{PR}$n after $\overline{MRS}$ goes LOW. For the whole RCLK cycle that the first quadrant is on $\overline{PAE}$n/$\overline{PR}$n the ESYNC ($\overline{PAE}$n/$\overline{PR}$n bus sync) output will be HIGH, for all other quadrants, this ESYNC output will be LOW. This ESYNC output provides the user with a mark with which they can synchronize to the $\overline{PAE}$n/$\overline{PR}$n bus, ESYNC is always HIGH for the RCLK cycle that the first quadrant of a device is present on the $\overline{PAE}$n/$\overline{PR}$n bus.

When devices are connected in expansion mode, only one device will be set as the Master, MAST input tied HIGH, all other devices will have MAST tied LOW. The master device is the first device to take control of the $\overline{PAE}$n/$\overline{PR}$n bus and will place its first quadrant on the bus on the rising edge of RCLK after the $\overline{MRS}$ input goes LOW. For the next 3 RCLK cycles the master device will maintain control of the $\overline{PAE}$n/$\overline{PR}$n bus and cycle its quadrants through it, all other devices hold their $\overline{PAE}$n/$\overline{PR}$n outputs in High-Impedance. When the master device has cycled all of its quadrants it passes a token to the next device in the chain and that device assumes control of the $\overline{PAE}$n/$\overline{PR}$n bus and then cycles its quadrants and so on, the $\overline{PAE}$n/$\overline{PR}$n bus control token being passed on from device to device. This token passing is done via the EXO outputs and EXI inputs of the devices ("$\overline{PAE}$ Expansion Out" and "$\overline{PAE}$ Expansion In"). The EXO output of the master device connects to the EXI of the second device in the chain and the EXO of the second connects to the EXI of the third and so on. The final device in a chain has its EXO connected to the EXI of the first device, so that once the $\overline{PAE}$n/$\overline{PR}$n bus has cycled through all quadrants of all devices, control of the $\overline{PAE}$n/$\overline{PR}$n will pass to the master device again and so on. The ESYNC of each respective device will operate independently and simply indicate when that respective device has taken control of the bus and is placing its first quadrant on to the $\overline{PAE}$n/$\overline{PR}$n bus.

When operating in single device mode the EXI input must be connected to the EXO output of the same device. In single device mode a token is still required to be passed into the device for accessing the $\overline{PAE}$n bus.

Please refer to Figure 33, *$\overline{PAE}$n/$\overline{PR}$n Bus – Polled Mode* for timing information.

ECHO READ CLOCK (ERCLK)

The Echo Read Clock output is provided in both HSTL and LVTTL mode, selectable via IOSEL. The ERCLK is a free-running clock output, it will always follow the RCLK input regardless of $\overline{REN}$ and RADEN.

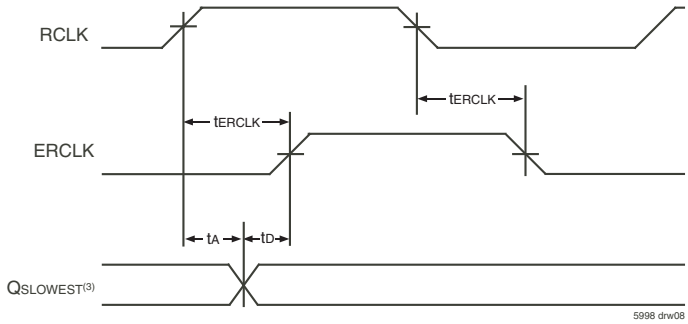
The ERCLK output follows the RCLK input with an associated delay. This delay provides the user with a more effective read clock source when reading data from the Qn outputs. This is especially helpful at high speeds when variables within the device may cause changes in the data access times. These variations in access time maybe caused by ambient temperature, supply voltage, device characteristics. The ERCLK output also compensates for any trace length delays between the Qn data outputs and receiving devices inputs.

Any variations effecting the data access time will also have a corresponding effect on the ERCLK output produced by the Queue device, therefore the ERCLK output level transitions should always be at the same position in time relative to the data outputs. Note, that ERCLK is guaranteed by design to be slower than the slowest Qn, data output. Refer to Figure 3, *Echo Read Clock and Data Output Relationship* and Figure 27, *Echo RCLK & Echo $\overline{REN}$ Operation* for timing information.

ECHO READ ENABLE ($\overline{EREN}$)

The Echo Read Enable output is provided in both HSTL and LVTTL mode, selectable via IOSEL.

The $\overline{EREN}$ output is provided to be used in conjunction with the ERCLK output and provides the reading device with a more effective scheme for reading data from the Qn output port at high speeds. The $\overline{EREN}$ output is controlled by internal logic that behaves as follows: The $\overline{EREN}$ output is active LOW for the RCLK cycle that a new word is read out of the Queue. That is, a rising edge of RCLK will cause $\overline{EREN}$ to go active (LOW) if $\overline{REN}$ is active and the Queue is NOT empty.



NOTES:
1. $\overline{REN}$ is LOW. $\overline{OE}$ is LOW.
2. $t_{ERCLK} > t_A$, guaranteed by design.
3. Qslowest is the data output with the slowest access time, t_A.
4. Time, t_D is greater than zero, guaranteed by design.

Figure 3. Echo Read Clock and Data Output Relationship

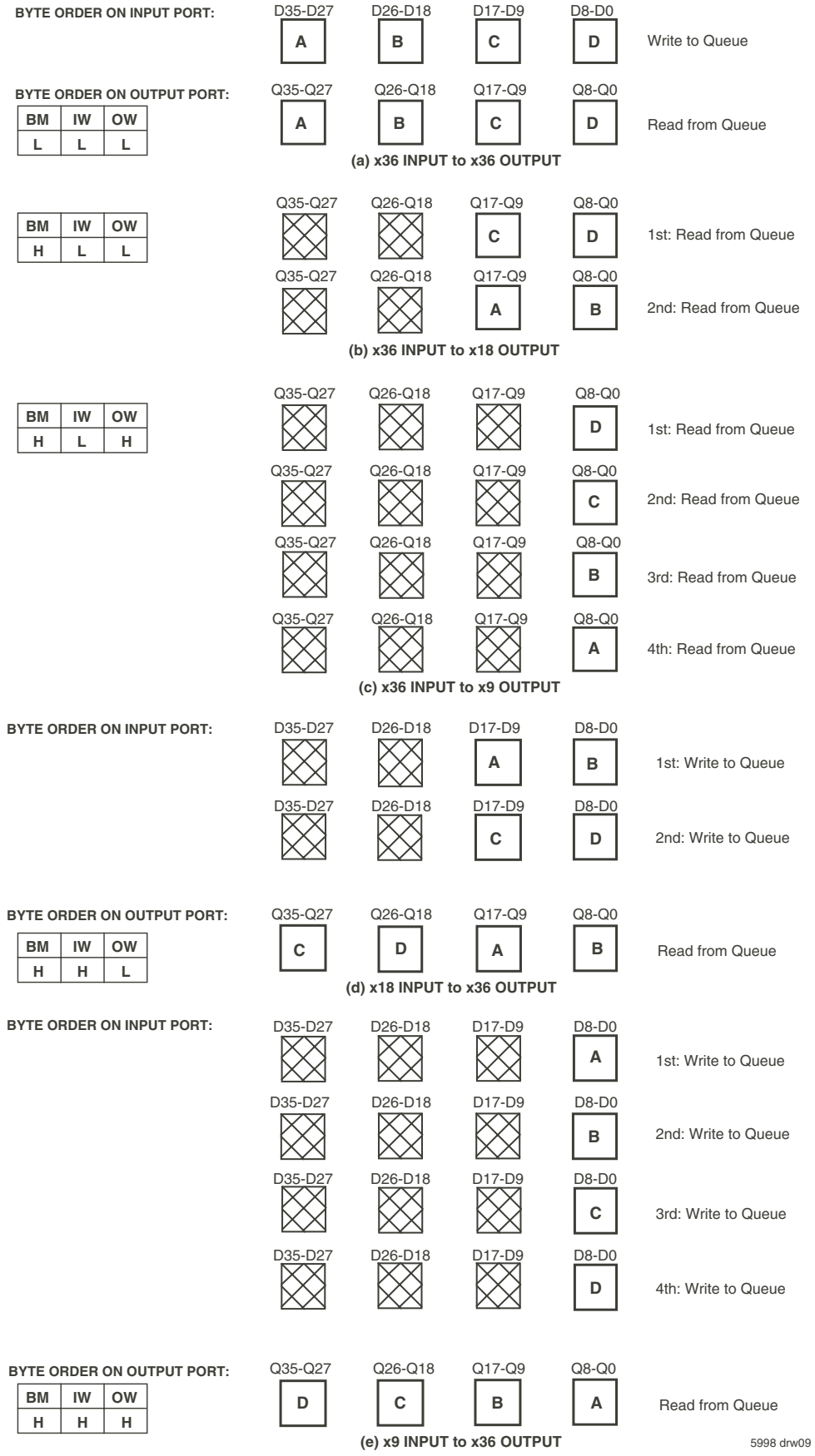
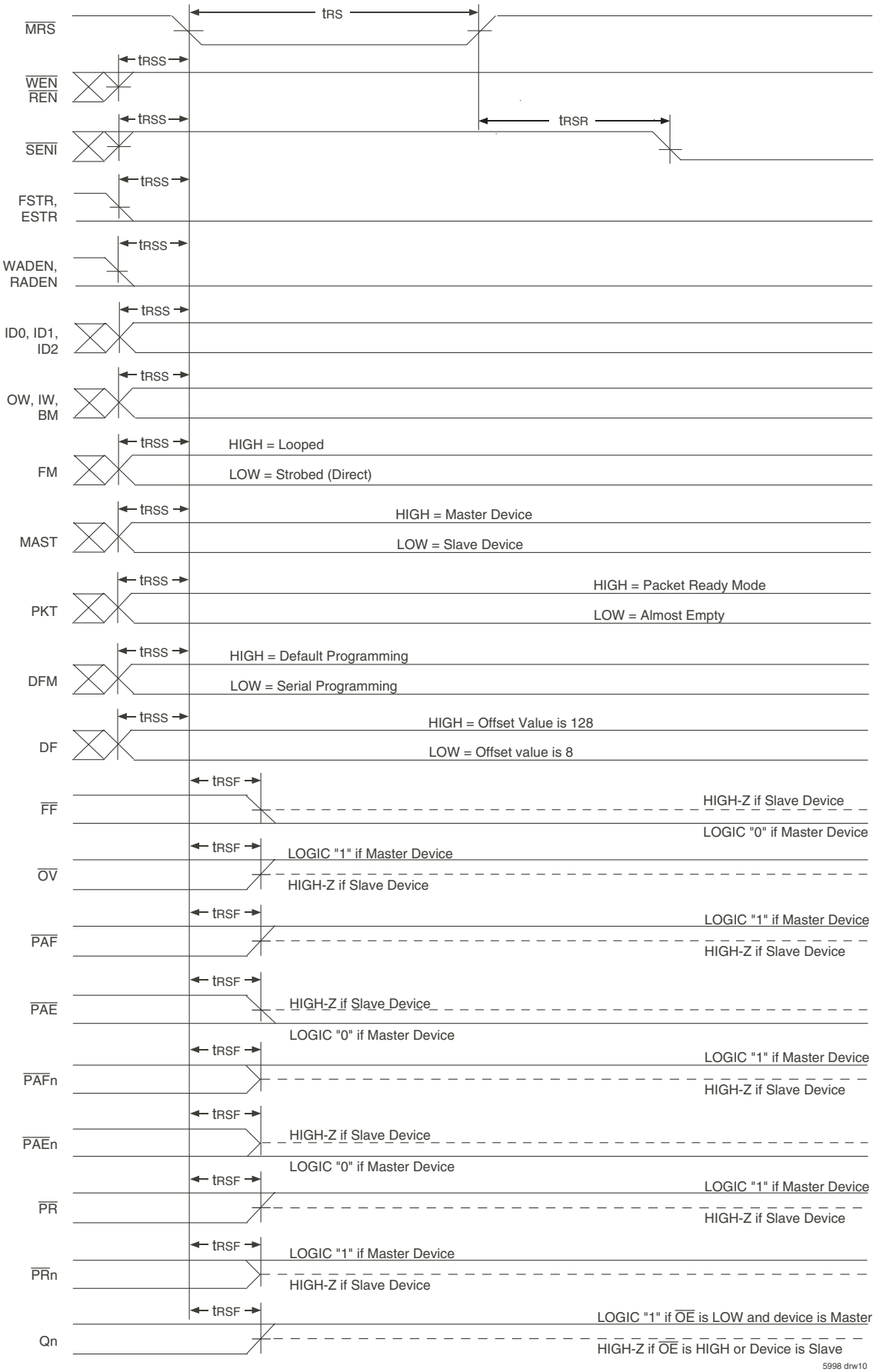
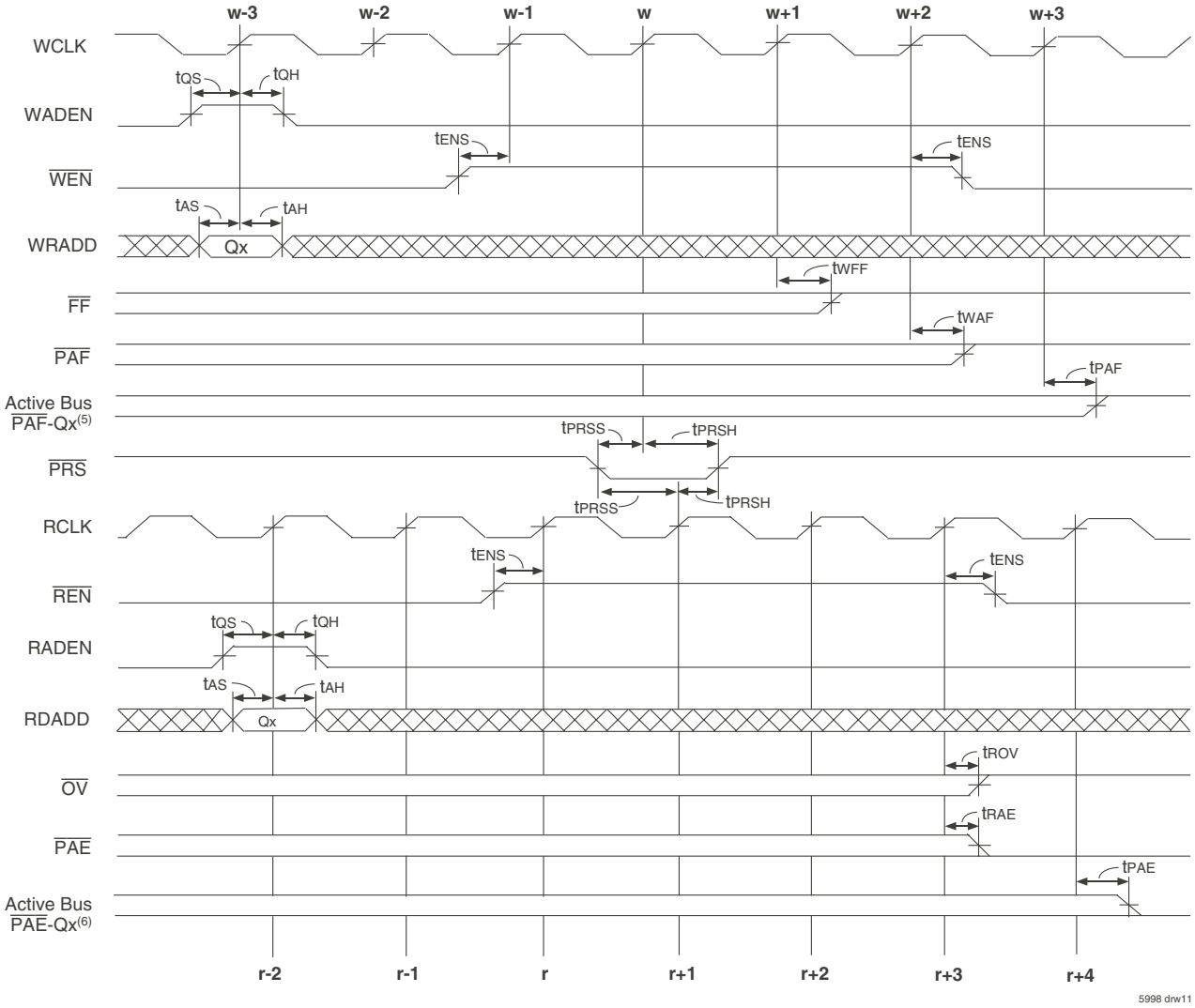


Figure 4. Bus-Matching Byte Arrangement



NOTES:
1. $\overline{OE}$ can toggle during this period.
2. $\overline{PRS}$ should be HIGH during a $\overline{MRS}$.

Figure 5. Master Reset



NOTES:
1. For a Partial Reset to be performed on a Queue, that Queue must be selected on both the write and read ports.
2. The queue must be selected a minimum of 3 clock cycles before the Partial Reset takes place, on both the write and read ports.
3. The Partial Reset must be LOW for a minimum of 1 WCLK and 1 RCLK cycle.
4. Writing or Reading to the queue (or a queue change) cannot occur until a minimum of 3 clock cycles after the Partial Reset has gone HIGH, on both the write and read ports.
5. The $\overline{PAF}$ flag output for Qx on the $\overline{PAF}$n flag bus may update one cycle later than the active $\overline{PAF}$ flag.
6. The $\overline{PAE}$ flag output for Qx on the $\overline{PAE}$n flag bus may update one cycle later than the active $\overline{PAE}$ flag.

Figure 6. Partial Reset

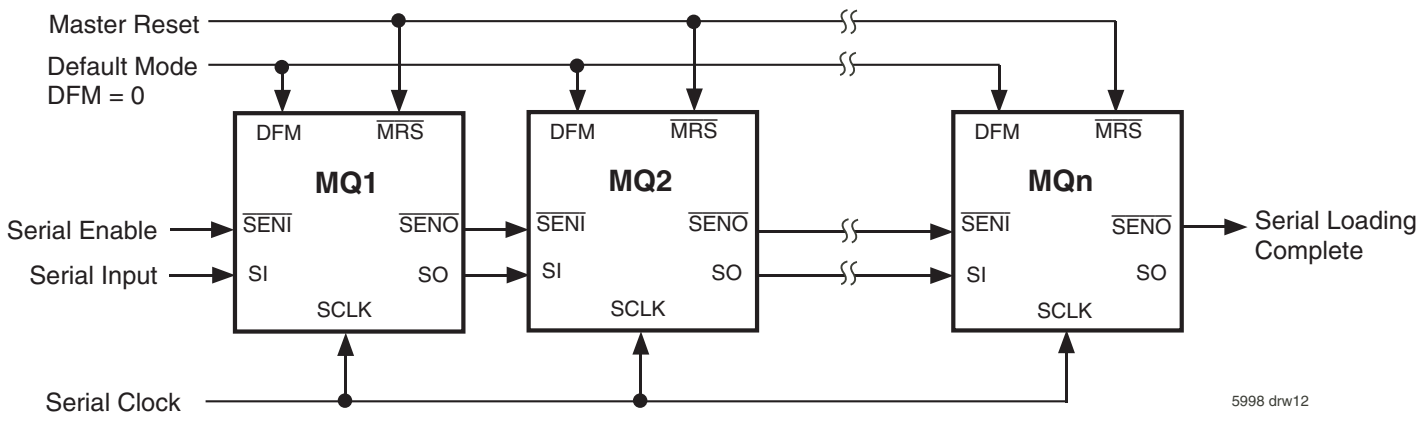
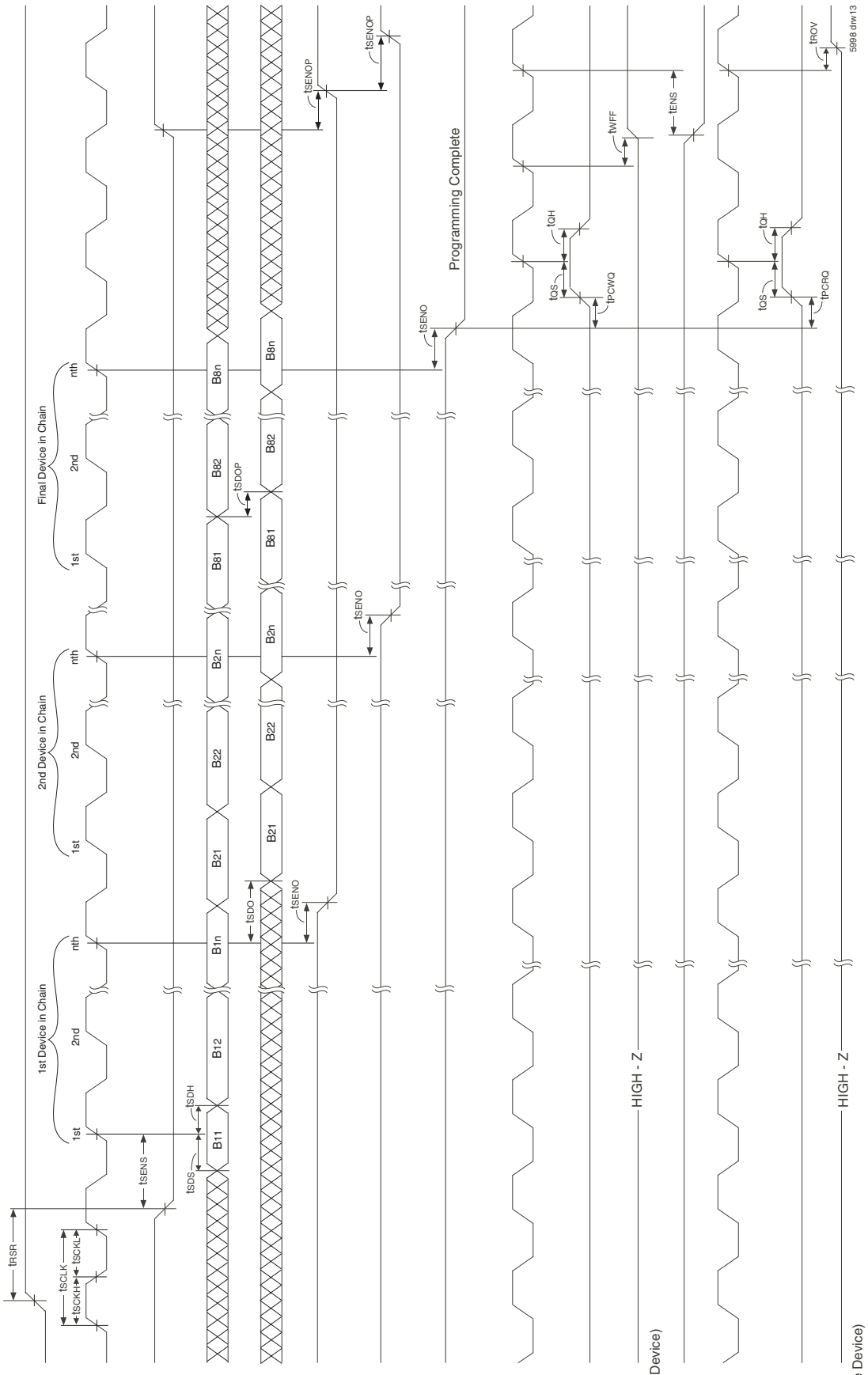
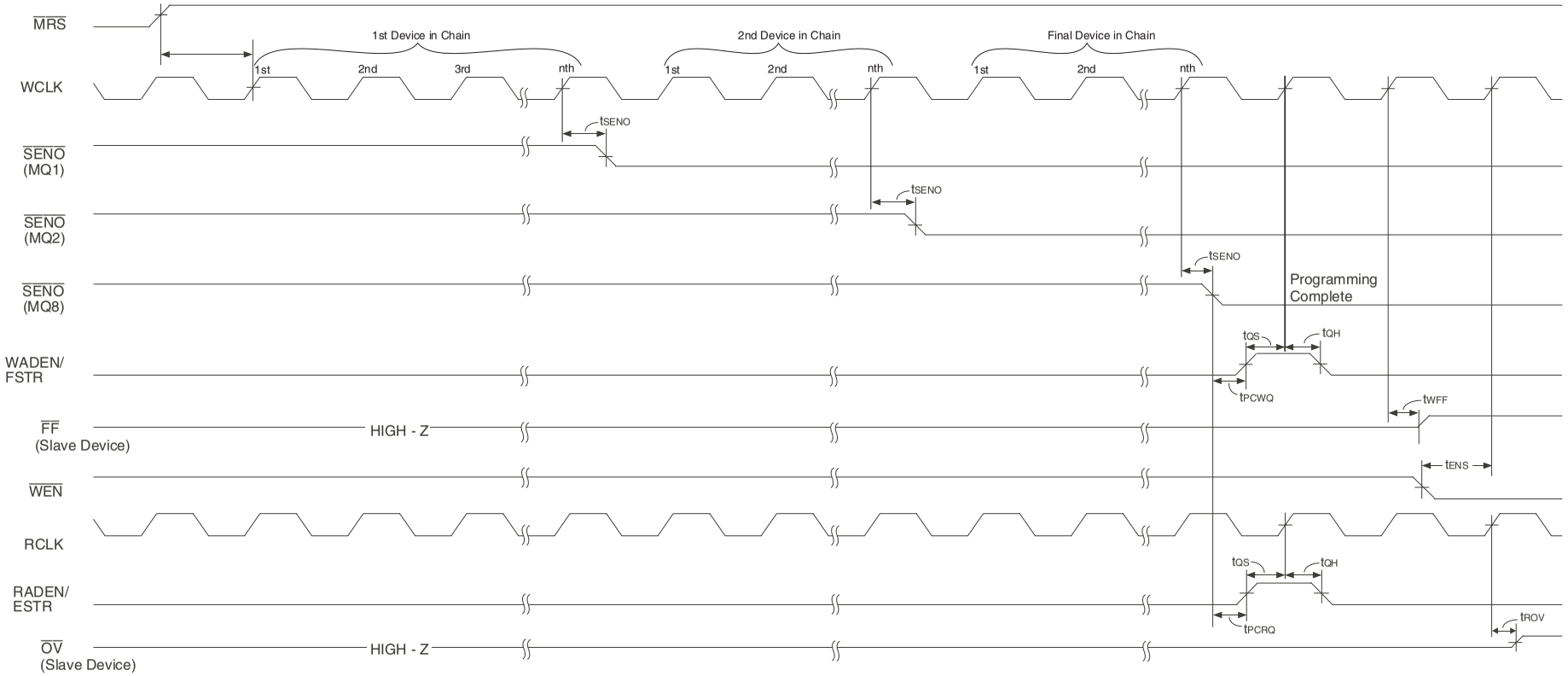


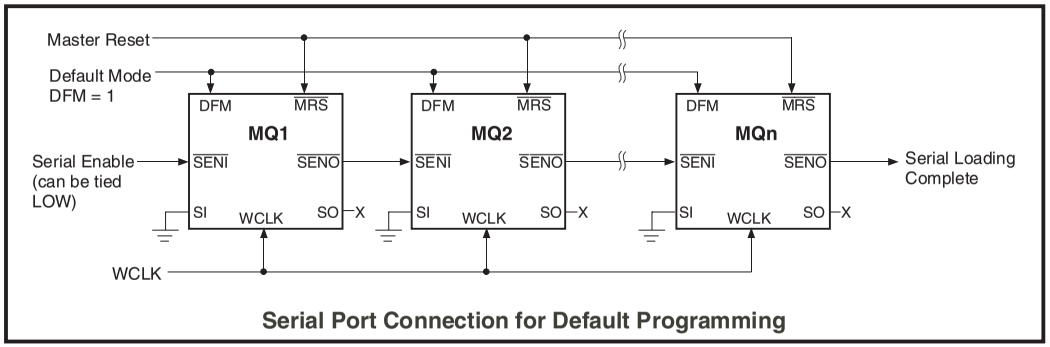
Figure 7. Serial Port Connection for Serial Programming



NOTES:
1. $\overline{SENI}$ can be toggled during serial loading. Once serial programming of a device is complete, the $\overline{SENI}$ and SI inputs become transparent. $\overline{SENI} \rightarrow \overline{SENO}$ and SI → SO.
2. DFM is LOW during Master Reset to provide Serial programming mode, DF is don't care.
3. When $\overline{SENO}$ of the final device is LOW no further serial loads will be accepted.
4. n = 19+(Qx72); where Q is the number of queues required for the IDT72T51546/72T51556.
5. This diagram illustrates 8 devices in expansion.
6. Programming of all devices must be complete ($\overline{SENO}$ of the final device is LOW), before any write or read port operations can take place, this includes queue selections.

Figure 8. Serial Programming



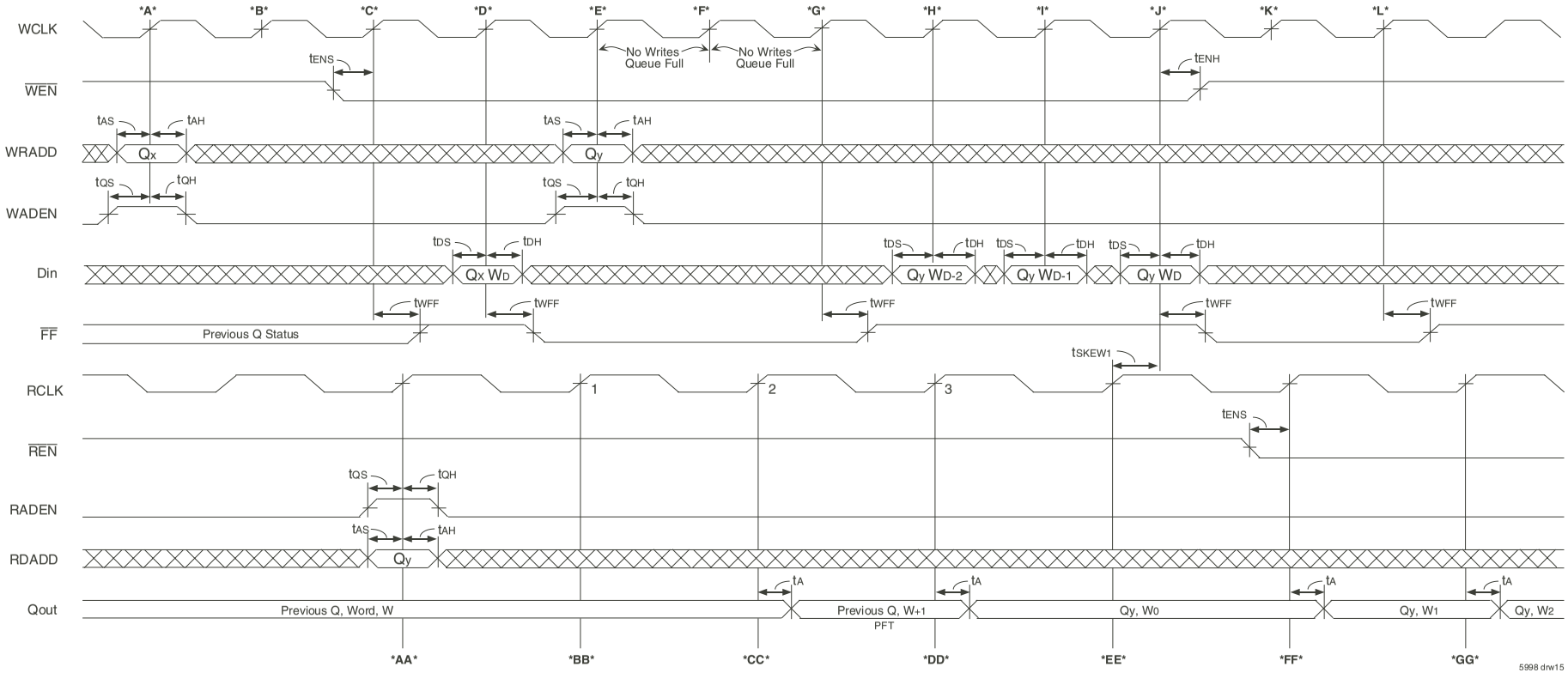


Serial Port Connection for Default Programming

5998 drw14

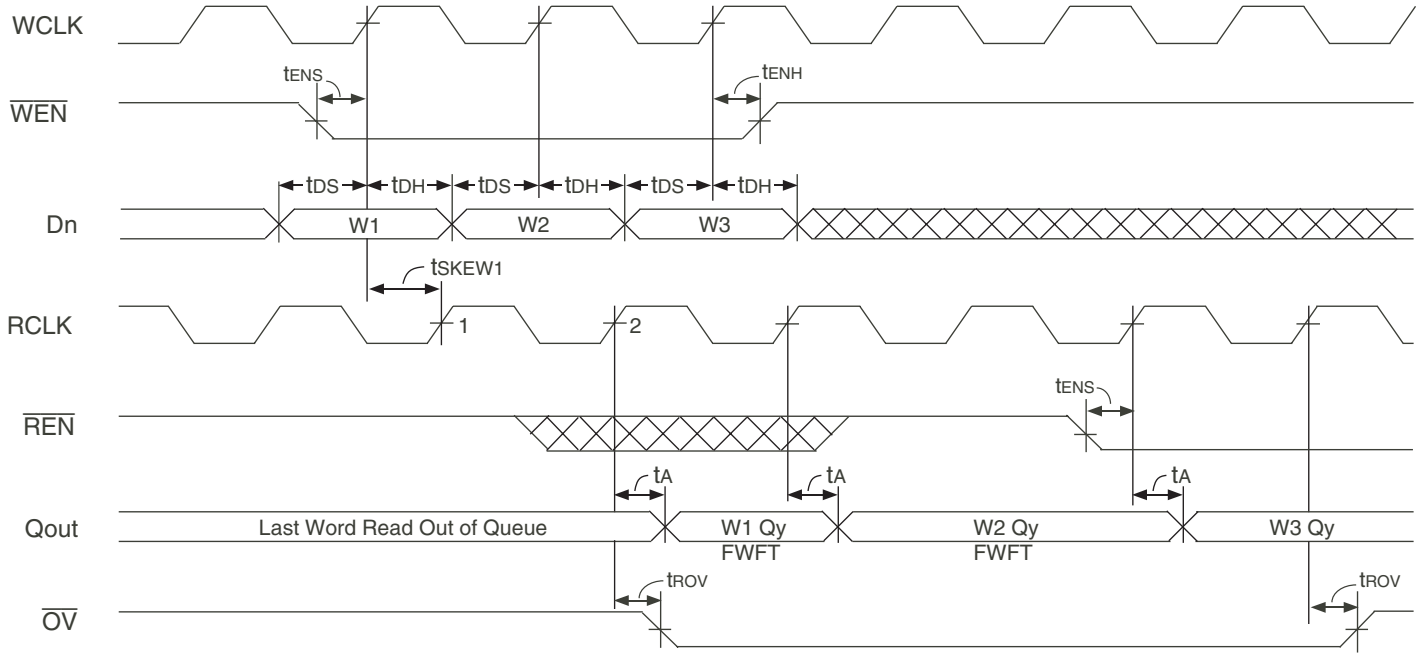
NOTES:
1. This diagram illustrates multiple devices connected in expansion. The $\overline{SENO}$ of the final device in a chain is the "programming complete" signal.
2. $\overline{SENI}$ of the first device in the chain can be held LOW
3. The $\overline{SENO}$ of a device should connect to the $\overline{SENI}$ of the next device in the chain. The final device $\overline{SENO}$ is used to indicate programming complete.
4. When Default Programming is complete the $\overline{SENO}$ of the final device will go LOW.
5. SCLK is not used and can be tied LOW.
6. Programming of all devices must be complete ($\overline{SENO}$ of the final device is LOW), before any write or read port operations can take place, this includes queue selections.

Figure 9. Default Programming



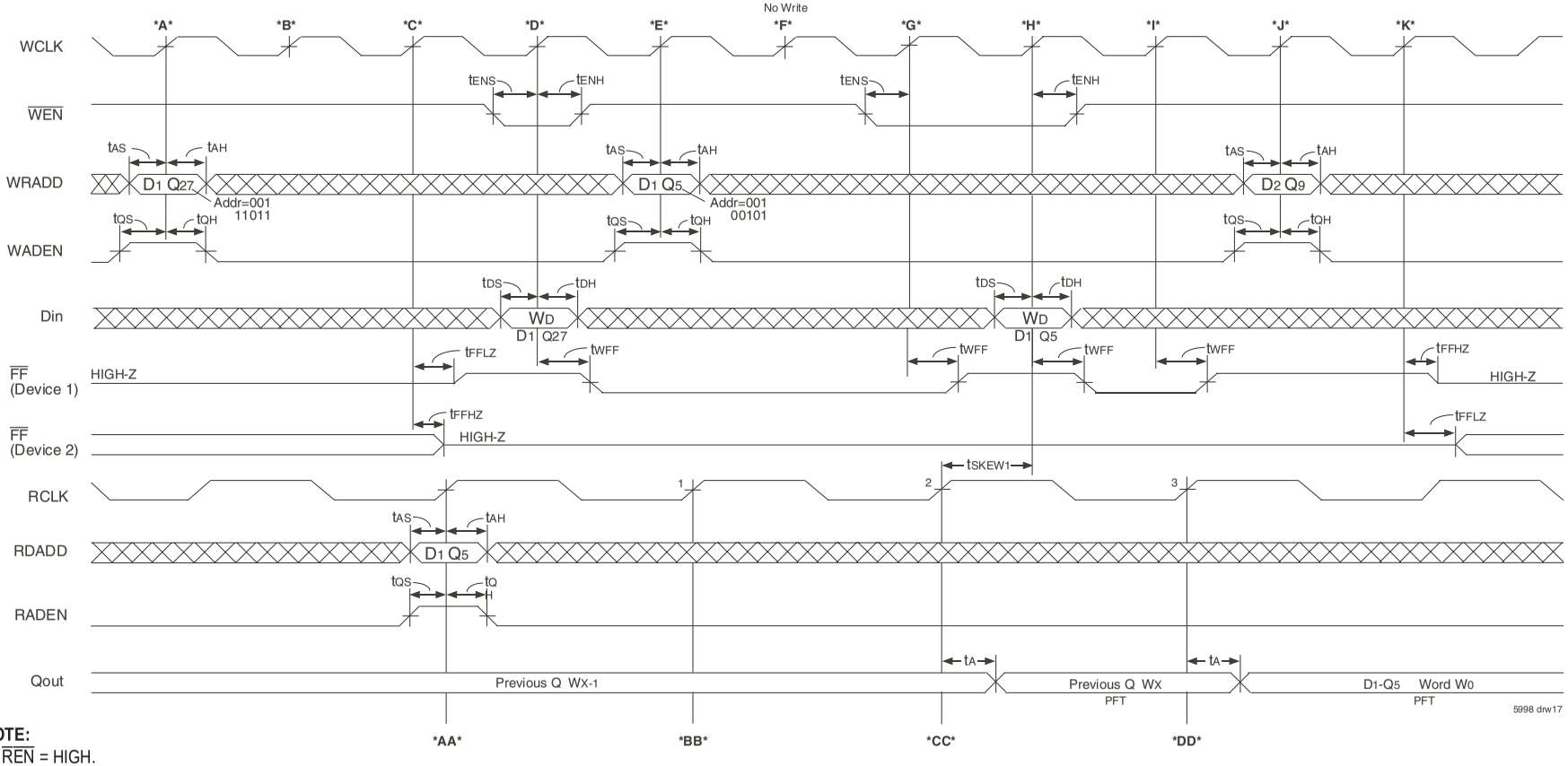
NOTE:
$\overline{OE}$ is active LOW.
Cycle:
A Queue, Qx is selected on the write port.
The $\overline{FF}$ flag is providing status of a previously selected queue, within the same device.
AA Queue, Qy is selected for read operations.
B The $\overline{FF}$ flag provides status of previous queue for 3 WCLK cycles.
BB Current word is kept on the output bus since $\overline{REN}$ is HIGH.
C The $\overline{FF}$ flag output updates to show the status of Qx, it is not full.
CC Word W+1 is read from the previous queue regardless of $\overline{REN}$ due to FWFT.
D Word, Wd is written into Qx. This causes Qx to go full.
DD The next available Word W0 of Qy is read out regardless of $\overline{REN}$, 3 RCLK cycles after queue selection. This is FWFT operation.
E Queue, Qy is selected within the same device as Qx. A write to Qx cannot occur on this cycle because it is full, $\overline{FF}$ is LOW.
EE No reads occur, $\overline{REN}$ is HIGH.
F Again, a write to Qx cannot occur on this cycle because it is full, $\overline{FF}$ is LOW.
FF Word, W1 is read from Qy, this causes Qy to go "not full", $\overline{FF}$ flag goes HIGH after time, $t_{SKEW1} + t_{WFF}$. Note, if t_{SKEW1} is violated the time $\overline{FF}$ HIGH will be: t_{SKEW1} + WCLK + t_{WFF}.
G The $\overline{FF}$ flag updates after time t_{WFF} to show that queue, Qy is not full.
GG Word, W2 is read from Qy.
H Word, Wd-2 is written into Qy.
I Word, Wd-1 is written into Qy.
J Word, Wd is written into Qy, this causes Qy to go full, $\overline{FF}$ goes LOW.
K A write to Qy cannot occur on this cycle because it is full, $\overline{FF}$ is LOW.
L Qy goes "not full" based on reading word W1 from Qy on cycle *FF*.

Figure 10. Write Queue Select, Write Operation and Full Flag Operation



NOTES:
1. Qy has previously been selected on both the write and read ports.
2. $\overline{OE}$ is LOW.
3. The First Word Latency = tSKEW1 + RCLK + tA. If tSKEW1 is violated an additional RCLK cycle must be added.

Figure 11. Write Operations & First Word Fall Through



NOTE:
1. $\overline{REN}$ = HIGH.

Cycle:

A Queue, Q27 of device 1 is selected on the write port.
The $\overline{FF}$ flag of device 1 is in High-Impedance, the write port of device 2 was previously selected.
$\overline{WEN}$ is HIGH so no write occurs.

AA Queue, Q5 of device 1 is selected on the read port.

B The $\overline{FF}$ flag stays in High-Impedance for 2 WCLK cycles.

BB Word, Wx-1 is held on the outputs for 2 RCLK cycles after a read Queue switch.

C The $\overline{FF}$ flag of device 2 goes to High-Impedance and the $\overline{FF}$ flag of device 1 goes to Low-Impedance, logic HIGH indicating that D1 Q27 is not full.
$\overline{WEN}$ is HIGH so no write occurs.

CC Word, Wx is read from the previously selected queue, (due to FWFT).

D Word, Wd is written into Q27 of D1. This write operation causes Q27 to go full, $\overline{FF}$ goes LOW.

DD The first word from Q5 of D1 selected on cycle *AA* is read out, this occurred regardless of $\overline{REN}$ due to FWFT. This read caused Q5 to go not full, therefore the $\overline{FF}$ flag will go HIGH after: t_{SKEW1} + t_{WFF}.
Note if t_{SKEW1} is violated the time to $\overline{FF}$ flag HIGH is t_{SKEW1} + WLCK + t_{WFF}.

E Queue, Q5 of device 1 is selected on the write port. No write occurs on this cycle.

F The $\overline{FF}$ flag stays in High-Impedance for 2 WCLK cycles.

G The $\overline{FF}$ flag updates to show the status of D1 Q5, it is not full, $\overline{FF}$ goes HIGH.

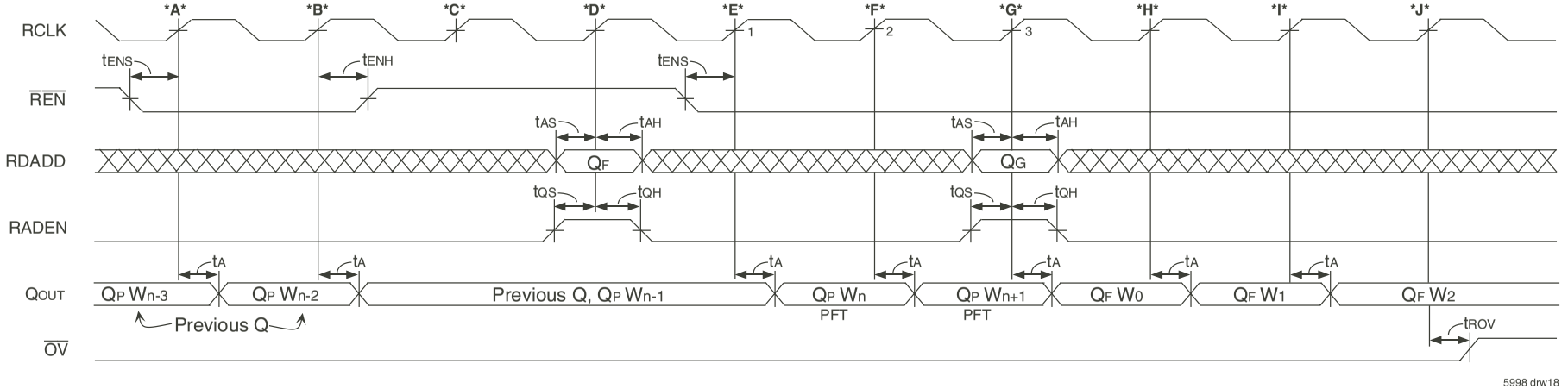
H Word, Wd is written into Q5 of D1. This causes the queue to go full, $\overline{FF}$ goes LOW.

I No write occurs regardless of $\overline{WEN}$, the $\overline{FF}$ flag is LOW preventing writes. The $\overline{FF}$ flag goes HIGH due to the read from Q5 of D1 on cycle *CC*. (This read is not an enabled read, it is due to the FWFT operation).

J Queue, Q9 of device 2 is selected on the write port.

K The $\overline{FF}$ flag of device 1 goes to High-Impedance, this device was deselected on the write port on cycle *I*. The $\overline{FF}$ flag of device 2 goes to Low-Impedance and provides status of Q9 of D2.

Figure 12. Full Flag Timing in Expansion Mode



Cycle:

A Word W_{n-3} is read from a previously selected queue Qp on the read port.

B W_{n-2} is read.

C Reads are disabled, W_{n-1} remains on the output bus.

D A new queue, Q_F is selected for read operations.

E Word W_{n-1} in Qp is read out.

F The next word available in current queue Q_P, W_{n+1} is read regardless of $\overline{REN}$ due to FWFT.

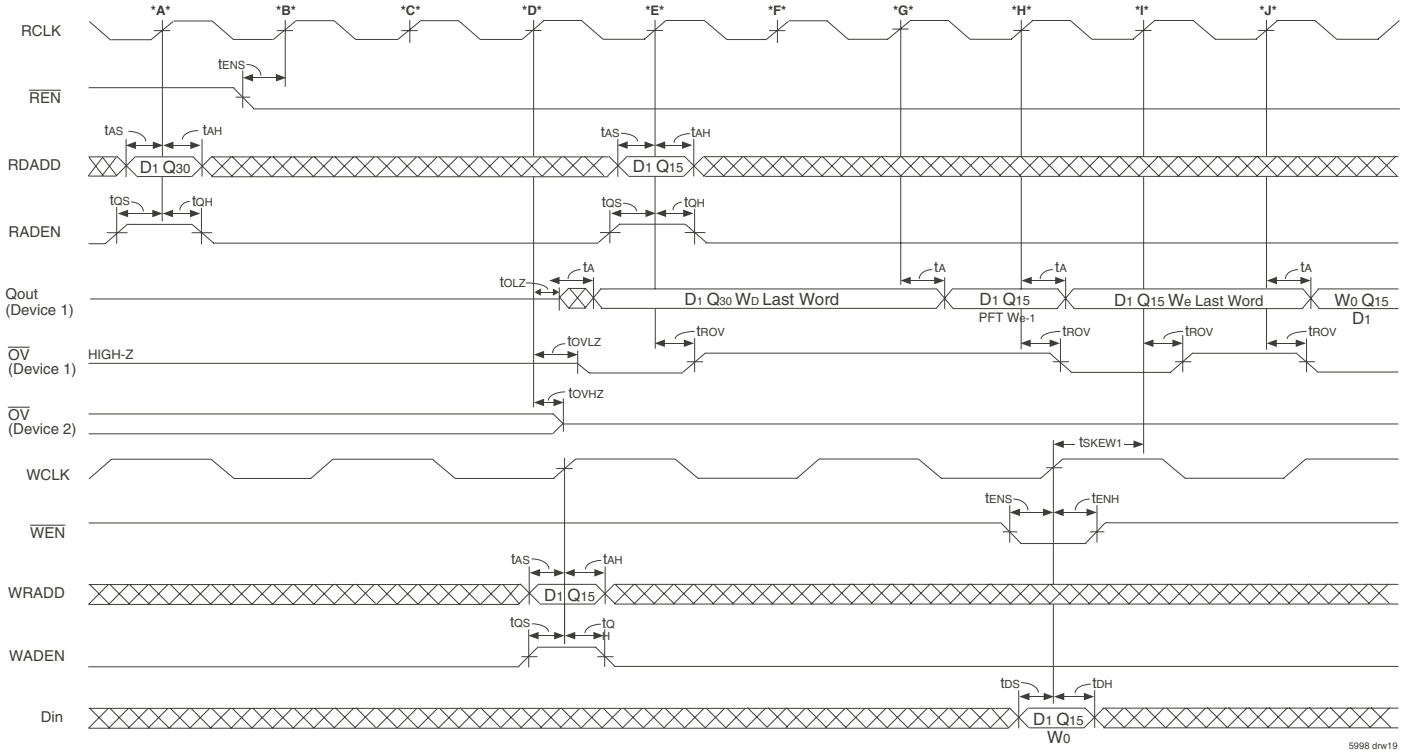
G The next word available in the new queue, Q_F-W_0 falls through to the output bus, again this is regardless of $\overline{REN}$. A new queue, Q_G is selected for read operations. (This queue is an empty queue).

H Word, W_1 is read from Q_F.

I Word, W_2 is read from Q_F.

J Word W_2 from Q_F remains on the output bus because Q_G is empty. The Output Valid Flag, $\overline{OV}$ goes HIGH to indicate that the current word is not valid, i.e. Q_G is empty. W_2 is the last word in Q_G.

Figure 13. Read Queue Select, Read Operation



Cycle:

A Queue 30 of Device 1 is selected for read operations. The $\overline{OV}$ is currently being driven by Device 2, a queue within device 2 is selected for reads. Device 2 also has control of Qout bus, its Qout outputs are in Low-Impedance. This diagram only shows the Qout outputs of device 1. (Reads are disabled).

B Reads are now enabled. A word from the previously selected queue of Device 2 will be read out.

C After a queue switch, there is a 3 RCLK latency for output data.

D The Qout of Device 1 goes to Low-Impedance and word Wd is read from Q30 of D1. This happens to be the last word of Q30. Device 2 places its Qout outputs into High-Impedance, device 1 has control of the Qout bus. The $\overline{OV}$ flag of Device 2 goes to High-Impedance and Device 1 takes control of $\overline{OV}$. The $\overline{OV}$ flag of Device 1 goes LOW to show that Wd of Q30 is valid.

E Queue 15 of device 1 is selected for read operations. The last word of Q30 was read on the previous cycle, therefore $\overline{OV}$ goes HIGH to indicate that the data on the Qout is not valid (Q30 was read to empty). Word, Wd remains on the output bus.

F The last word of Q30 remains on the Qout bus, $\overline{OV}$ is HIGH, indicating that this word has been previously read.

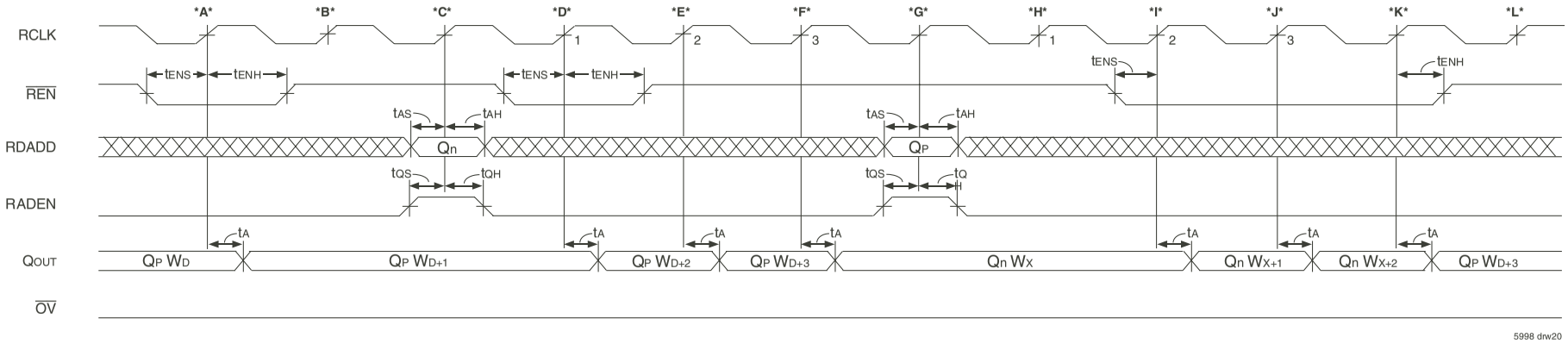
G The next word (We-1), available from the newly selected queue, Q15 of device 1 is now read out. This will occur regardless of $\overline{REN}$, 2 RCLK cycles after queue selection due to the FWFT operation. The $\overline{OV}$ flag updates 3 RCLK cycles after a queue selection.

H The last word, We is read from Q15, this queue is now empty.

I The $\overline{OV}$ flag goes HIGH to indicate that Q15 was read to empty on the previous cycle.

J Due to a write operation the $\overline{OV}$ flag goes LOW and data word W0 is read from Q15. The latency is: t_{SKEW1} + 1*RCLK + t_{ROV}.

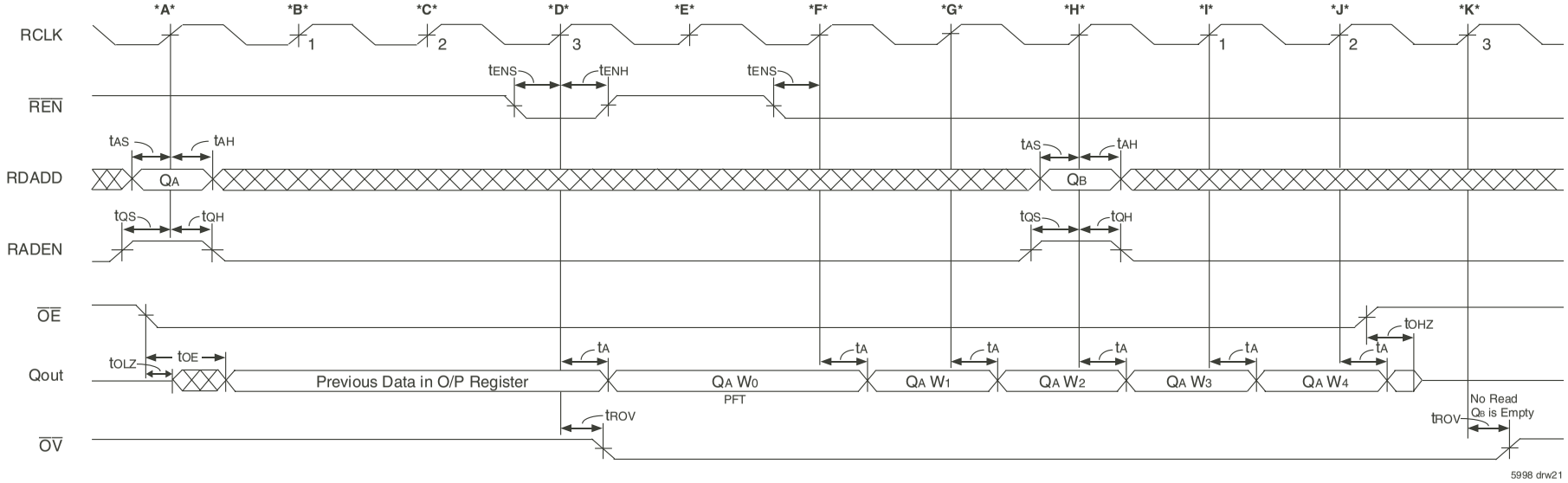
Figure 14. Output Valid Flag Timing (In Expansion Mode)



Cycle:

A Word Wd+1 is read from the previously selected queue, Qp.
B Reads are disabled, word Wd+1 remains on the output bus.
C A new queue, Qn is selected for read port operations.
D Word, WD+2 of Qp is read out.
E Word WD+3 of Qp is read out regardless of $\overline{REN}$ due to FWFT operation.
F The next available word Wx of Qn is read out regardless of $\overline{REN}$, 3 RCLK cycles after queue selection. This is FWFT operation.
G The queue, Qp is again selected.
H Current Word is kept on the output bus since $\overline{REN}$ is HIGH.
I Word Wx+2 is read from Qn. This is read out regardless of $\overline{REN}$ due to FWFT operation.
J Word WD+3 is read from Qp.
K Word WD+4 is read from Qp.
L Reads are disabled on this cycle, therefore no further reads occur.

Figure 15. Read Queue Selection with Reads Disabled



NOTES:

1. The Output Valid flag, $\overline{OV}$ is HIGH therefore the previously selected queue has been read to empty. The Output Enable input is Asynchronous, therefore the Qout output bus will go to Low-Impedance after time t_{OLZ}. The data currently on the output register will be available on the output after time t_{OE}. This data is the previous data on the output register, this is the last word read out of the previous queue.
2. In expansion mode the $\overline{OE}$ inputs of all devices should be connected together. This allows the output busses of all devices to be High-Impedance controlled.

Cycle:

A Queue A is selected for reads. No data will fall through on this cycle, the previous queue was read to empty.
B No data will fall through on this cycle, the previous queue was read to empty.
C Previous data kept on output bus since there is no read operation.
D Word, W0 from Q_A is read out regardless of $\overline{REN}$ due to FWFT operation. The $\overline{OV}$ flag goes LOW indicating that Word W0 is valid.
E Reads are disabled therefore word, W0 of Q_A remains on the output bus.
F Reads are again enabled so word W1 is read from Q_A.
G Word W2 is read from Q_A.
H Queue, Q_B is selected on the read port. This queue is actually empty. Word, W3 is read from Q_A.
I Word, W4 is read from Q_A.
J Output Enable is taken HIGH, this is Asynchronous so the output bus goes to High-Impedance after time, t_{OHZ}.
K Output Valid flag, $\overline{OV}$ goes HIGH to indicate that Q_B is empty. Data on the output port is no longer valid.

Figure 16. Read Queue Select, Read Operation and $\overline{OE}$ Timing

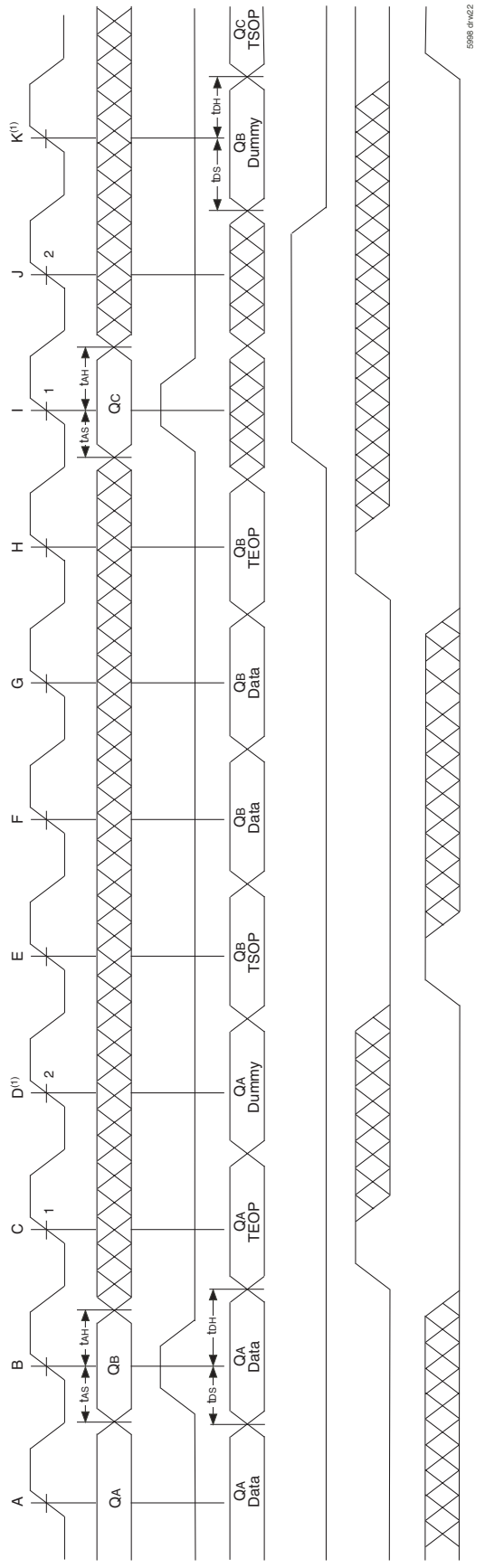
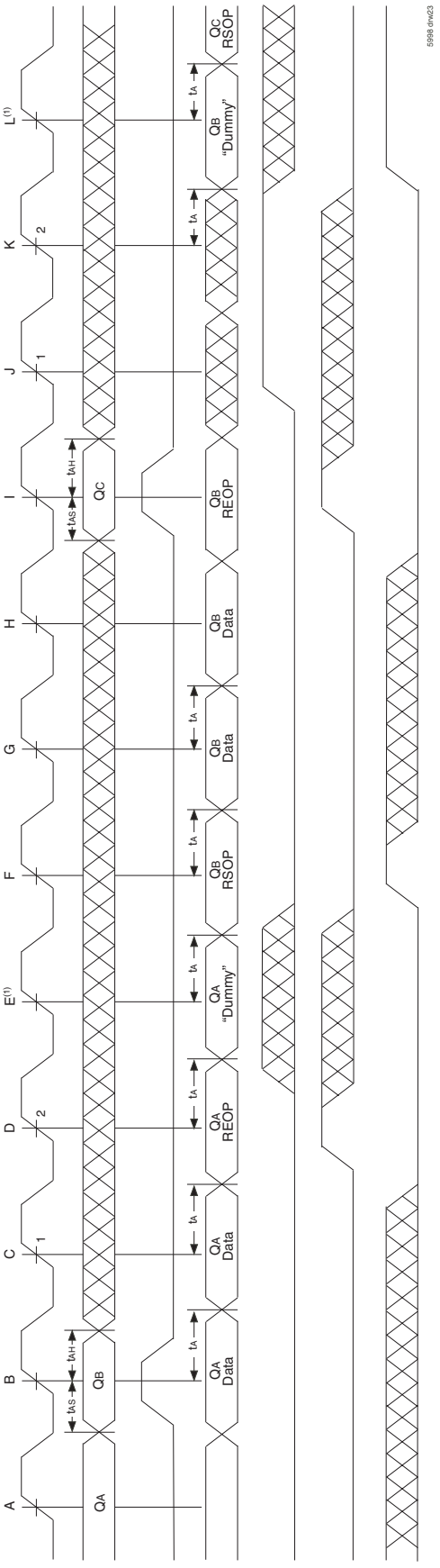


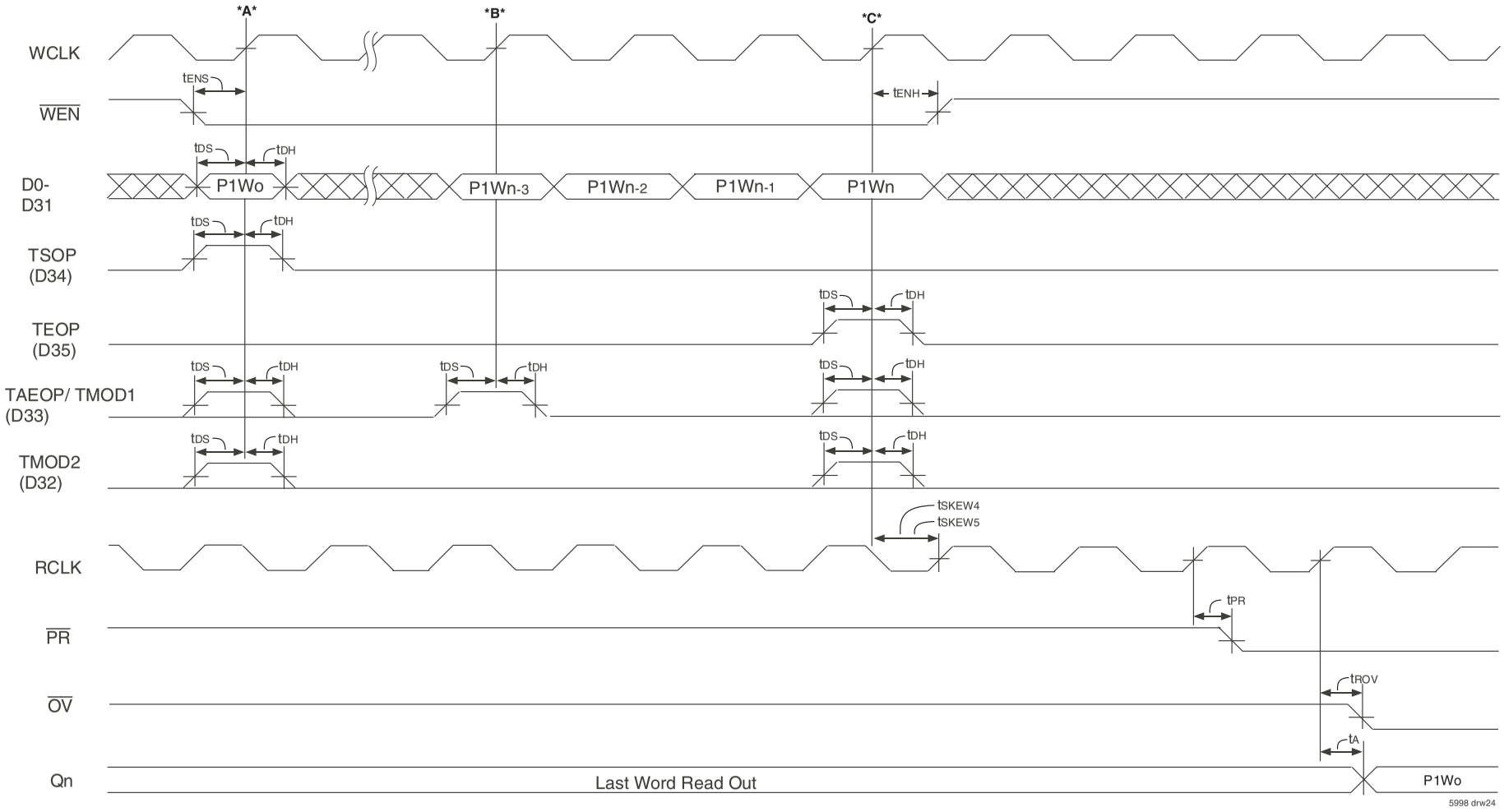
Figure 17. Writing in Packet Mode during a Queue change

NOTE:
1. Do not Write an SOP or EOP on "D" or "K". A filler word is needed.



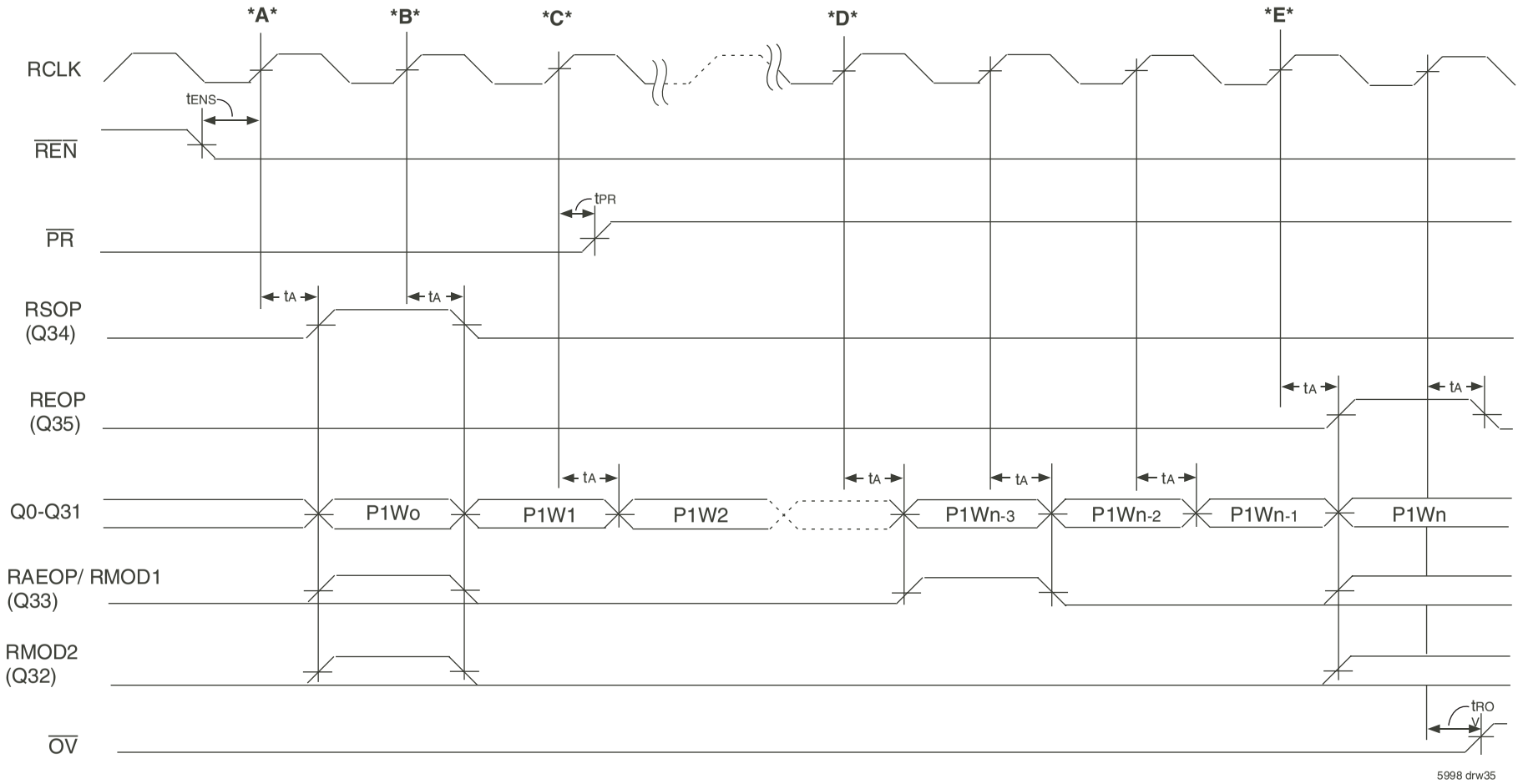
NOTE:
1. Do not Read an SOP or EOP on "E" or "L". A filler word is needed.

Figure 18. Reading in Packet Mode during a Queue change



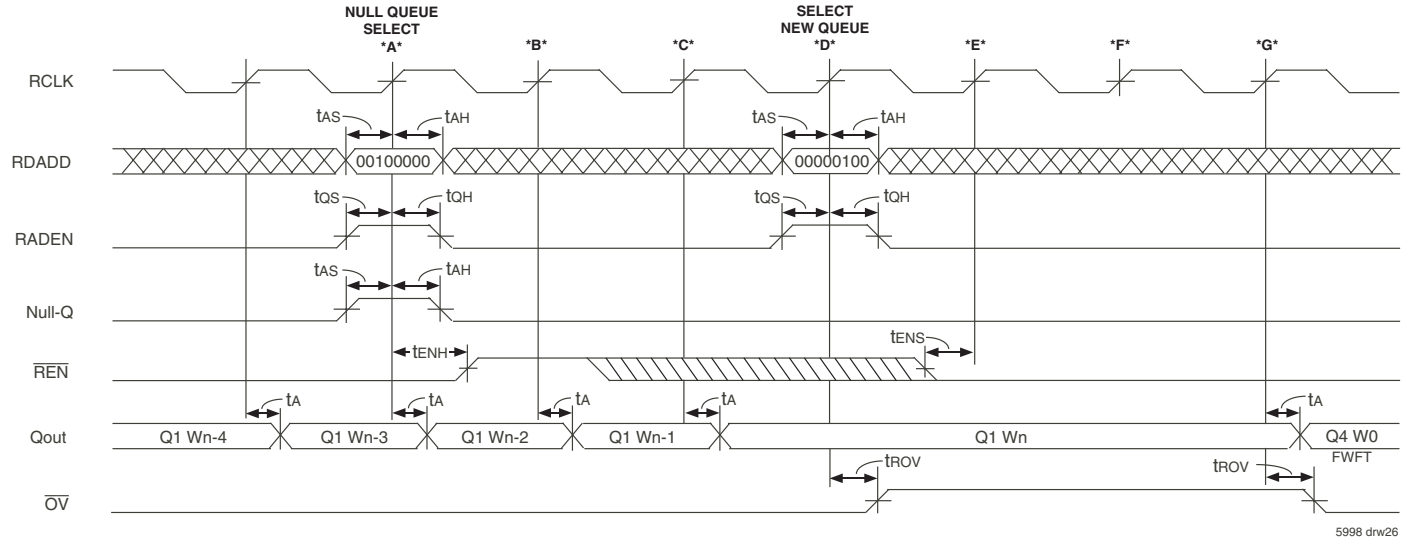
NOTES:
1. $\overline{REN}$ is HIGH.
2. If tSKEW4 is violated $\overline{PR}$ may take one additional RCLK cycle.
3. If tSKEW5 is violated the $\overline{OV}$ may take one additional RCLK cycle.
4. $\overline{PR}$ will always go LOW on the same cycle or 1 cycle ahead of $\overline{OV}$ going LOW, (assuming the last word of the packet is the last word in the queue).
5. In Packet mode, words cannot be read from a queue until a complete packet has been written into that queue, regardless of $\overline{REN}$.

Figure 19. Data Input (Transmit) Packet Mode of Operation



NOTE:
1. In Packet mode, words cannot be read from a queue until a complete packet has been written into that queue, regardless of $\overline{REN}$.
2. The $\overline{PR}$ flag will go HIGH on cycle *C* regardless of $\overline{REN}$.
3. The $\overline{OV}$ flag will go HIGH (preventing further reads), when the last complete packet has been read out. If there is a partial packet (an incomplete packet) in the queue the $\overline{OV}$ flag will remain HIGH until further writes have completed the packet.

Figure 20. Data Output (Receive) Packet Mode of Operation



NOTES:

1. The purpose of the Null queue operation is so that the user can stop reading a block (packet) of data from a queue without filling the 2 stage output pipeline with the next words from that queue.
2. Please see Figure 22, *Null Queue Flow Diagram*.

Cycle:

A Null Q of device 0 is selected, when word Wn-3 from previously selected Q1 is read.

C $\overline{REN}$ is HIGH and Wn (Last Word of the Packet) of Q1 is pipelined onto the O/P register.

Note: *B* and *C* are a minimum 3 RCLK cycles between queue selects.

D The Null Q is seen as an empty Queue on the read side, therefore Wn of Q1 remains in the O/P register and $\overline{OV}$ goes HIGH. A new queue, Q4 is selected.

G 1st word, W0 of Q4 falls through present on the O/P register after 3 RCLK cycles after the queue select.

Figure 21. Read Operation and Null Queue Select

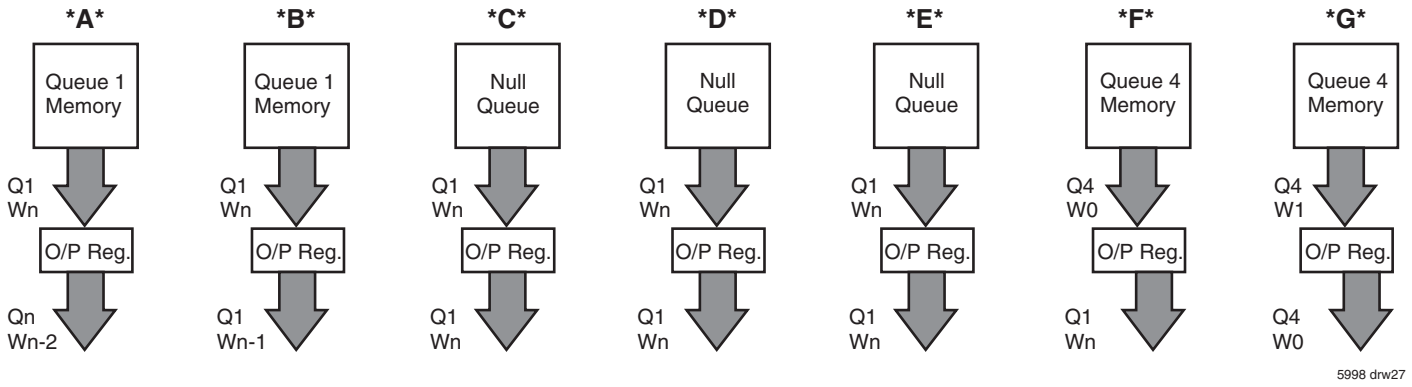
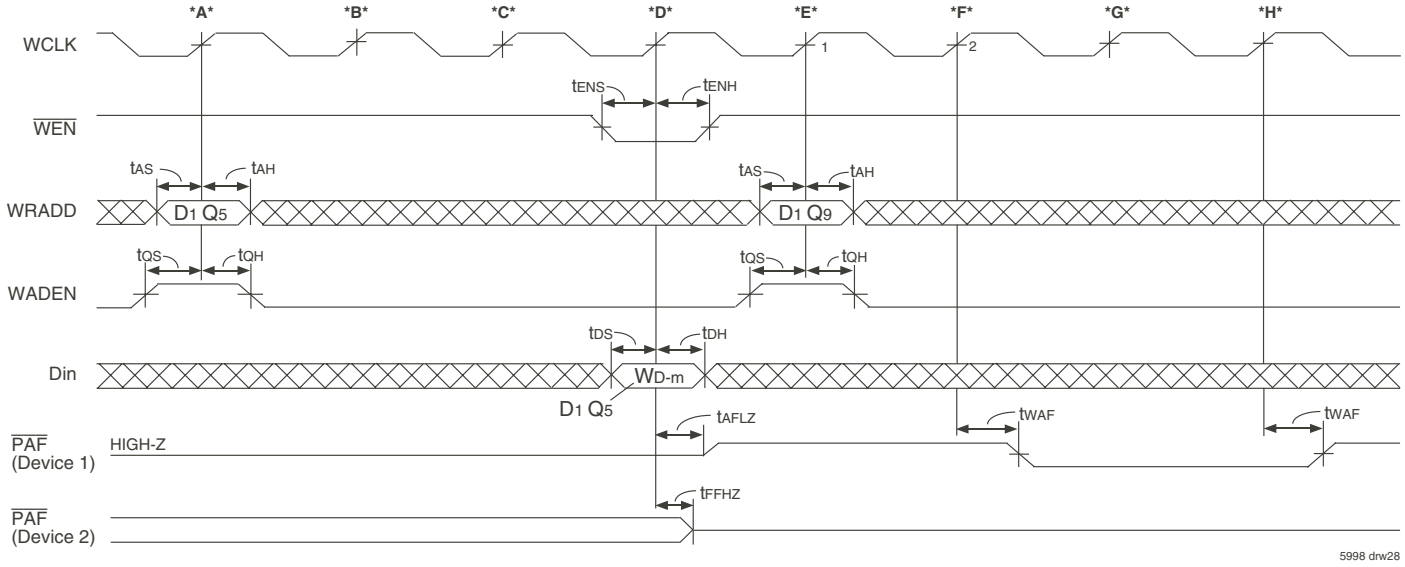
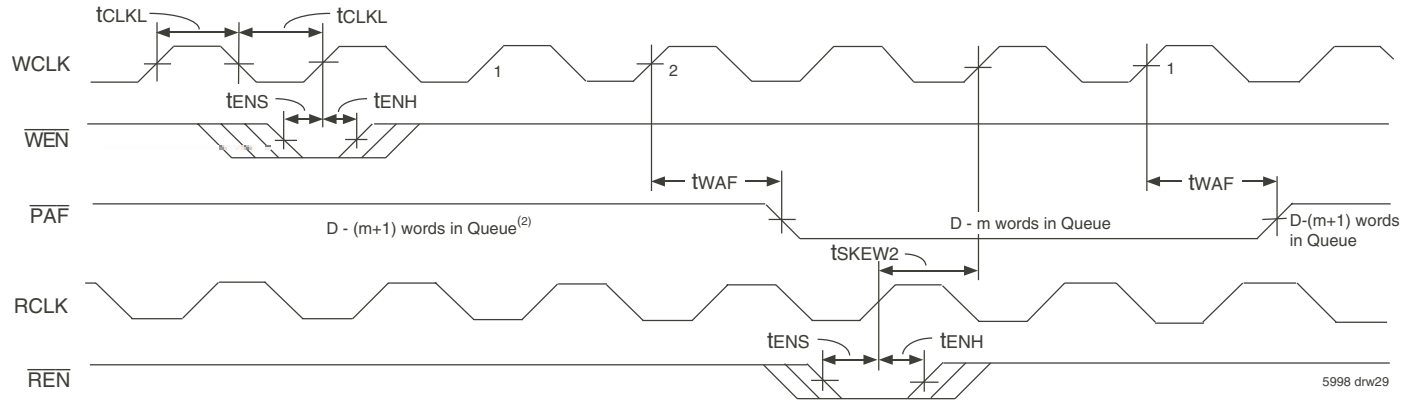


Figure 22. Null Queue Flow Diagram



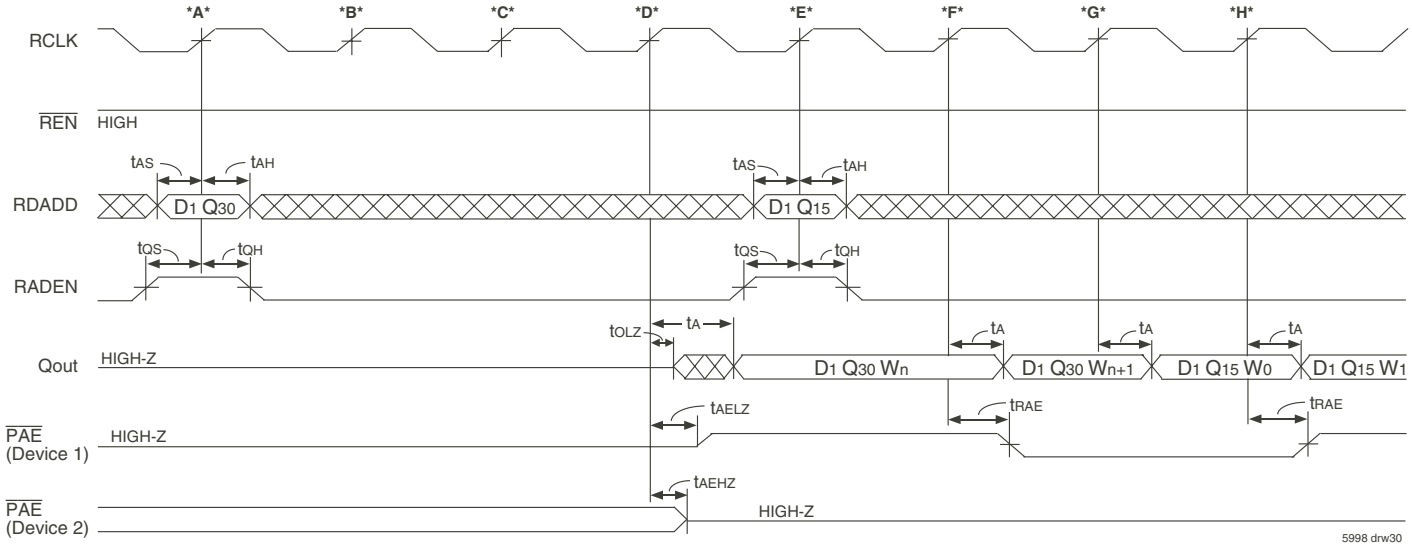
Cycle:
A Queue 5 of Device 1 is selected on the write port. A queue within Device 2 had previously been selected. The $\overline{PAF}$ output of device 1 is High-Impedance.
B No write occurs.
C No write occurs.
D Word, Wd-m is written into Q5 causing the $\overline{PAF}$ flag to go from LOW to HIGH. The flag latency is 3 WCLK cycles + t_{WAF}.
E Queue 9 in device 1 is now selected for write operations. This queue is not almost full, therefore the $\overline{PAF}$ flag will update after a 3 WCLK + t_{WAF} latency.
F The $\overline{PAF}$ flag goes LOW based on the write 2 cycles earlier.
G No write occurs.
H The $\overline{PAF}$ flag goes HIGH due to the queue switch to Q9.

Figure 23. Almost Full Flag Timing and Queue Switch



NOTE:
1. The waveform here shows the $\overline{PAF}$ flag operation when no queue switches are occurring and a queue selected on both the write and read ports is being written to then read from at the almost full boundary.
   Flag Latencies:
   Assertion: 2*WCLK + t_{WAF}
   De-assertion: t_{SKEW2} + WCLK + t_{WAF}
   If t_{SKEW2} is violated there will be one extra WCLK cycle.

Figure 24. Almost Full Flag Timing



Cycle:

A Queue 30 of Device 1 is selected on the read port. A queue within Device 2 had previously been selected. The $\overline{PAE}$ flag output and the data outputs of device 1 are High-Impedance.

B No read occurs.

C No read occurs.

D The $\overline{PAE}$ flag output now switches to device 1. Word, Wn is read from Q30 due to the FWFT operation. This read operation from Q30 is at the almost empty boundary, therefore $\overline{PAE}$ will go LOW 2 RCLK cycles later.

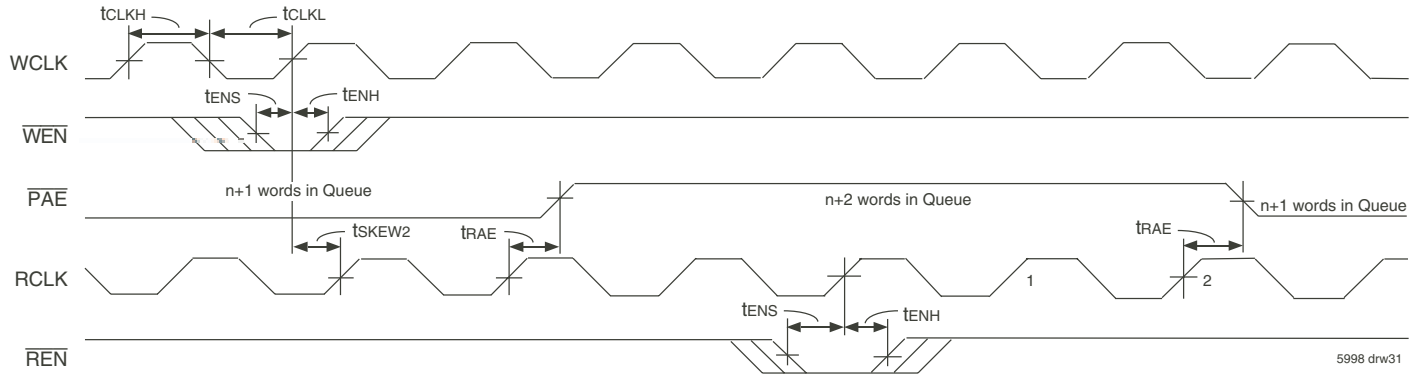
E Q15 of device 1 is selected.

F The $\overline{PAE}$ flag goes LOW due to the read from Q30 2 RCLK cycles earlier. Word Wn+1 is read out due to the FWFT operation.

G Word, W0 is read from Q15 due to the FWFT operation.

H The $\overline{PAE}$ flag goes HIGH to show that Q15 is not almost empty.

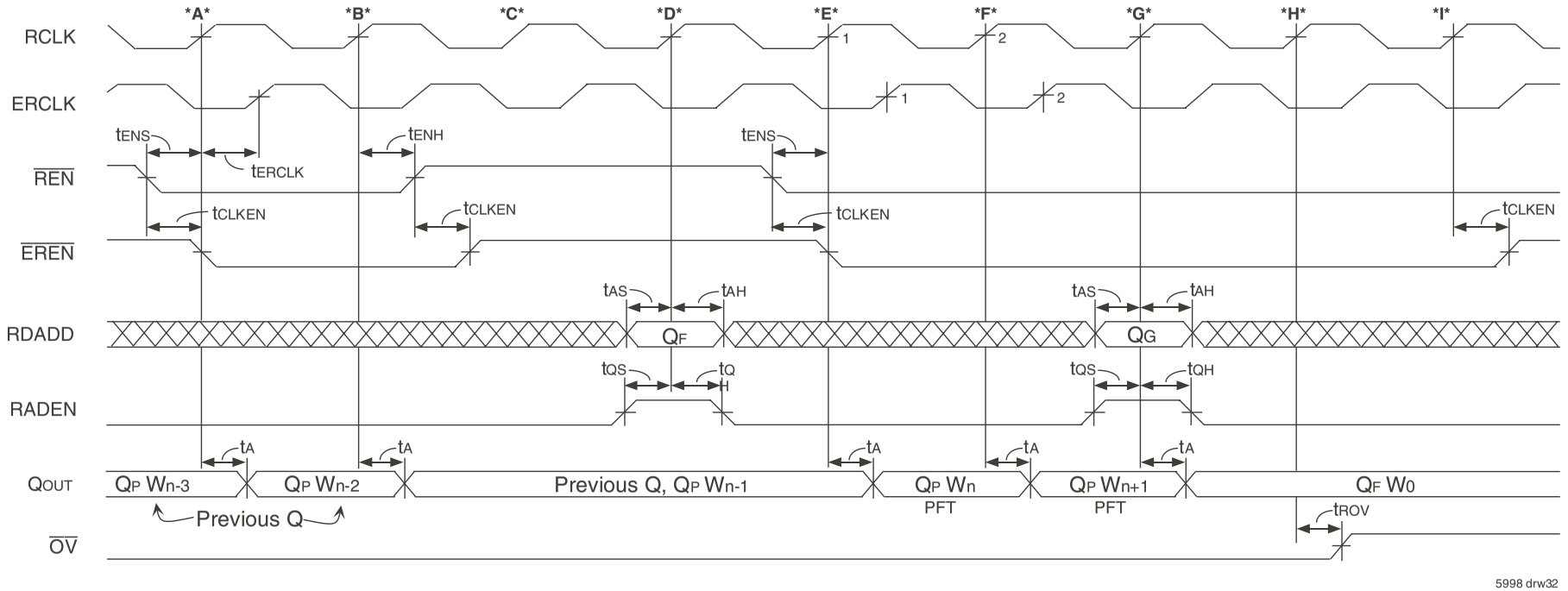
Figure 25. Almost Empty Flag Timing and Queue Switch



NOTE:

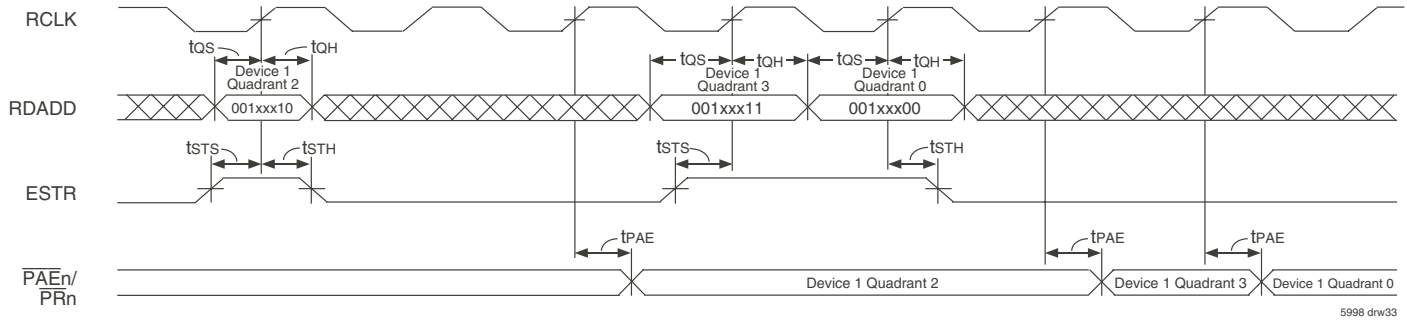
1. The waveform here shows the $\overline{PAE}$ flag operation when no queue switches are occurring and a queue selected on both the write and read ports is being written to then read from at the almost empty boundary.
   Flag Latencies:
   Assertion: 2*RCLK + t_{RAE}
   De-assertion: t_{SKEW2} + RCLK + t_{RAE}
   If t_{SKEW2} is violated there will be one extra RCLK cycle.

Figure 26. Almost Empty Flag Timing



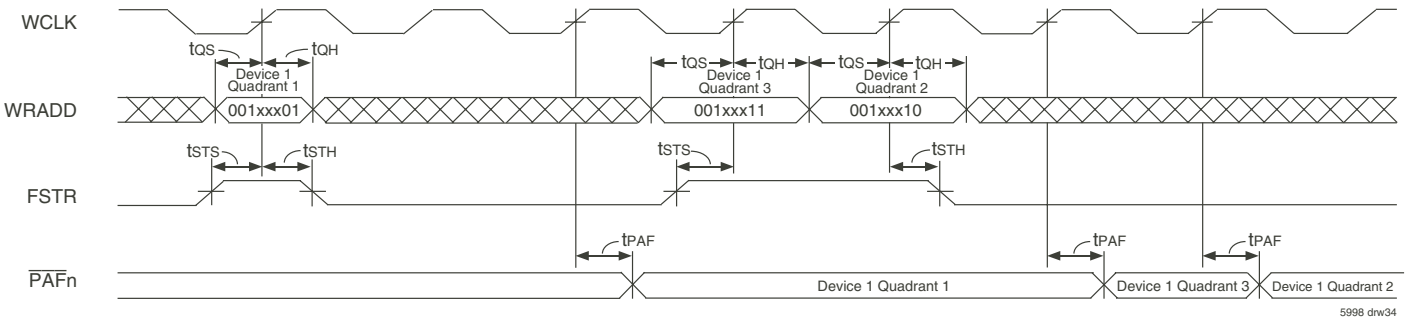
Cycle:
A $\overline{EREN}$ follow $\overline{REN}$ provided that the current queue (Qp) is not empty.
B $\overline{EREN}$ stays active since a new word (W_{n-1}) from Qp is placed on the output bus.
C, D $\overline{EREN}$ goes HIGH since no new word has been placed on the output bus on this cycle.
E $\overline{REN}$ goes LOW, new word placed on output bus, so $\overline{EREN}$ goes LOW.
F, G $\overline{EREN}$ stays active since a new word from Qp has been placed on the output bus.
H W_0 is the last word in Q_F thus $\overline{OV}$ goes HIGH.
I $\overline{EREN}$ goes HIGH since no new word has been placed on the output bus and Q_F is empty.

Figure 27. Echo RCLK and ECHO $\overline{REN}$ Operation



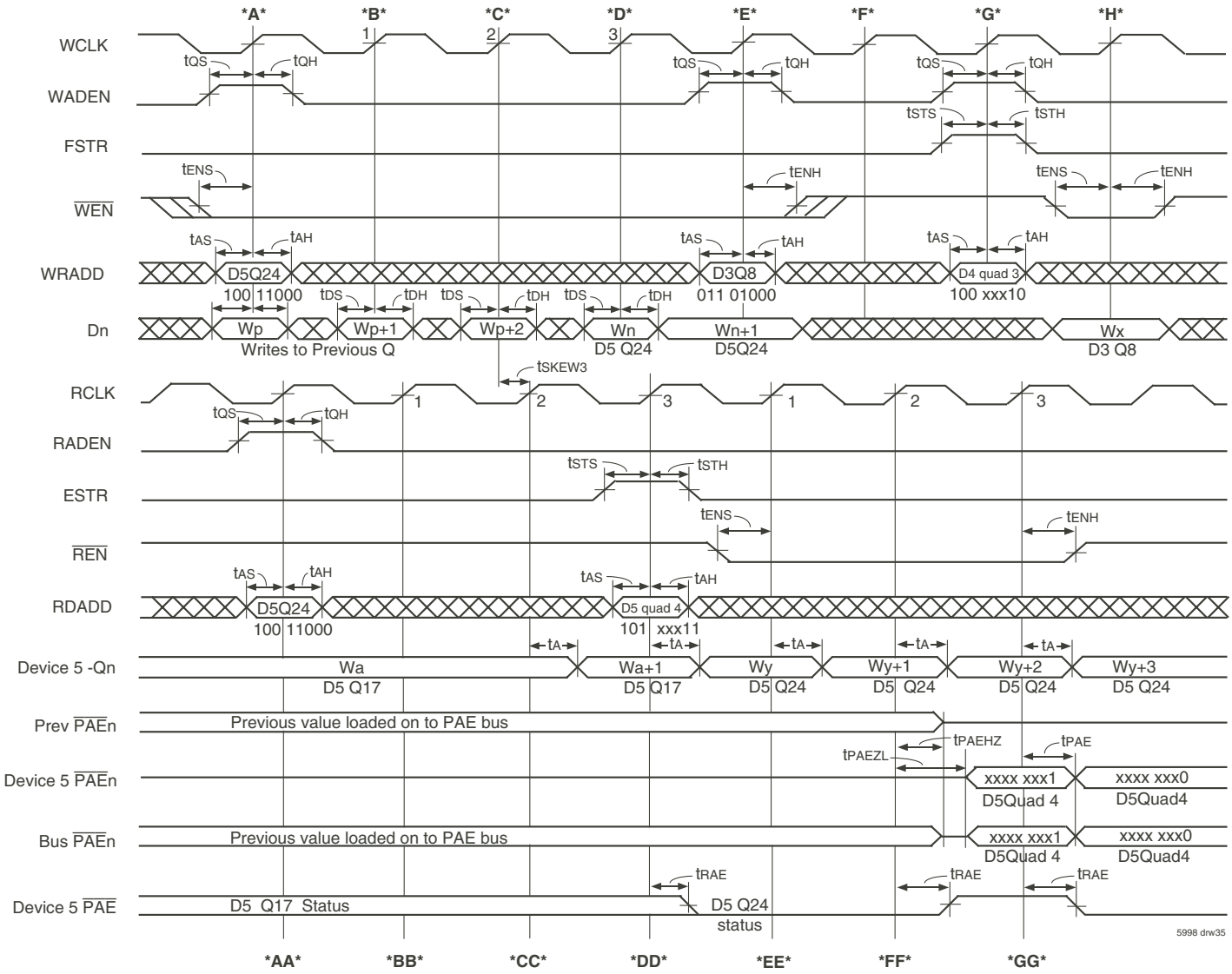
NOTES:
1. Quadrants can be selected on consecutive cycles.
2. On an RCLK cycle that the ESTR is HIGH, the RADEN input must be LOW.
3. There is a latency of 2 RCLK for the $\overline{PAE}$n bus to switch.

Figure 28. $\overline{PAE}$n/$\overline{PR}$n - Direct Mode - Quadrant Selection



NOTES:
1. Quadrants can be selected on consecutive cycles.
2. On a WCLK cycle that the FSTR is HIGH, the WADEN input must be LOW.
3. There is a latency of 2 WCLK for the $\overline{PAF}$n bus to switch.

Figure 29. $\overline{PAF}$n - Direct Mode - Quadrant Selection



Cycle:

A Queue 24 of Device 5 is selected for write operations.
Word, Wp is written into the previously selected queue.

AA Queue 24 of Device 5 is selected for read operations.
A quadrant from another device has control of the $\overline{PAE}$n bus.
The discrete $\overline{PAE}$ output of device 5 is currently in High-Impedance and the $\overline{PAE}$ active flag is controlled by the previously selected device.

B Word Wp+1 is written into the previously selected queue.

BB Current Word is kept on the output bus since $\overline{REN}$ is HIGH.

C Word Wp+2 is written into the previously selected queue.

CC Word Wa+1 of D5 Q17 is read due to FWFT.

D Word, Wn is written into the newly selected queue, Q24 of D5. This write will cause the $\overline{PAE}$ flag on the read port to go from LOW to HIGH (not almost empty) after time, t_{SKEW3} + RCLK + t_{RAE} (if t_{SKEW3} is violated one extra RCLK cycle will be added).

DD Word, Wy from the newly selected queue, Q24 will be read out due to FWFT operation.
Quadrant 4 of Device 5 is selected on the $\overline{PAE}$n bus. Q24 of device 5 will therefore have is $\overline{PAE}$ status output on $\overline{PAE}$[0]. There is a single RCLK cycle latency before the $\overline{PAE}$n bus changes to the new selection.

E Queue 8 of Device 3 is selected for write operations.
Word Wn+1 is written into Q24 of D5.

EE Word, Wy+1 is read from Q24 of D5.

F No writes occur.

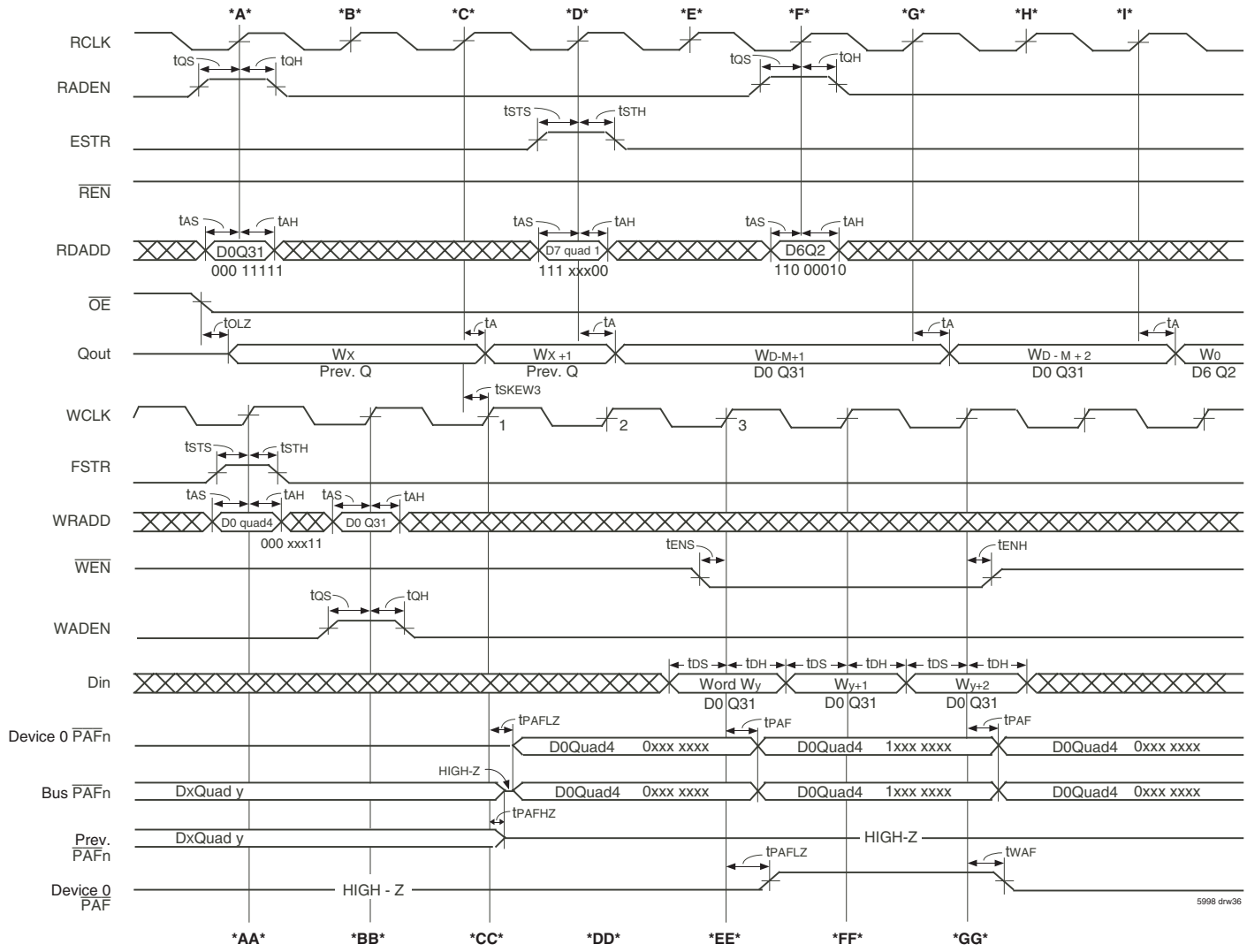
FF Word, Wy+2 is read from Q24 of D5.
The $\overline{PAE}$n bus changes control to D5, the $\overline{PAE}$n outputs of D5 go to Low-Impedance and quadrant 4 is placed onto the outputs. The device of the previously selected quadrant now places its $\overline{PAE}$n outputs into High-Impedance to prevent bus contention.
The discrete $\overline{PAE}$ flag will go HIGH to show that Q24 of D5 is not almost empty. Q24 of device 5 will have its $\overline{PAE}$ status output on $\overline{PAE}$[0].

G Quadrant 3 of device 4 is selected on the write port for the $\overline{PAF}$n bus.

GG The $\overline{PAE}$n bus updates to show that Q24 of D5 is almost empty based on the reading out of word, Wy+1.
The discrete $\overline{PAE}$ flag goes LOW to show that Q24 of D5 is almost empty based on the reading of Wy+1.

H Word, Wx is written into Q8 of D3.

Figure 30. $\overline{PAE}$n - Direct Mode, Flag Operation



Cycle:

A Queue 31 of device 0 is selected for read operations.
The last word in the output register is available on Qout. $\overline{OE}$ was previously taken LOW so the output bus is in Low-Impedance.

AA Quadrant 4 of device 0 is selected for the $\overline{PAF}$n bus. The bus is currently providing status of a previously selected quadrant, Quad Y of device X.

B No read operation.

BB Queue 31 of device 0 is selected on the write port.

C Word, Wx+1 is read out from the previous queue due to the FWFT effect.

CC $\overline{PAF}$n continues to show status of Quad4 D0.
The $\overline{PAF}$n bus is updated with the quadrant selected on the previous cycle, D0 Quad 4. $\overline{PAF}$[7] is LOW showing the status of queue 31.
The $\overline{PAF}$n outputs of the device previously selected on the $\overline{PAF}$n bus go to High-Impedance.

D A new quadrant, Quad 1 of Device 7 is selected for the $\overline{PAF}$n bus.
Word, Wd-m+1 is read from Q31 D0 due to the FWFT operation. This read is at the $\overline{PAF}$n boundary of queue D0 Q31. This read will cause the $\overline{PAF}$[7] output to go from LOW to HIGH (almost full to not almost full), after a delay t_{SKEW3} + WCLK + tPAF. If t_{SKEW3} is violated add an extra WCLK cycle.

DD No write operation.

E No read operations occur, $\overline{REN}$ is HIGH.

EE $\overline{PAF}$[7] goes HIGH to show that D0 Q31 is not almost empty due to the read on cycle *C*.
The active queue $\overline{PAF}$ flag of device 0 goes from High-Impedance to Low-Impedance.
Word, Wy is written into D0 Q31.

F Queue 2 of Device 6 is selected for read operations.

FF Word, Wy+1 is written into D0 Q31.

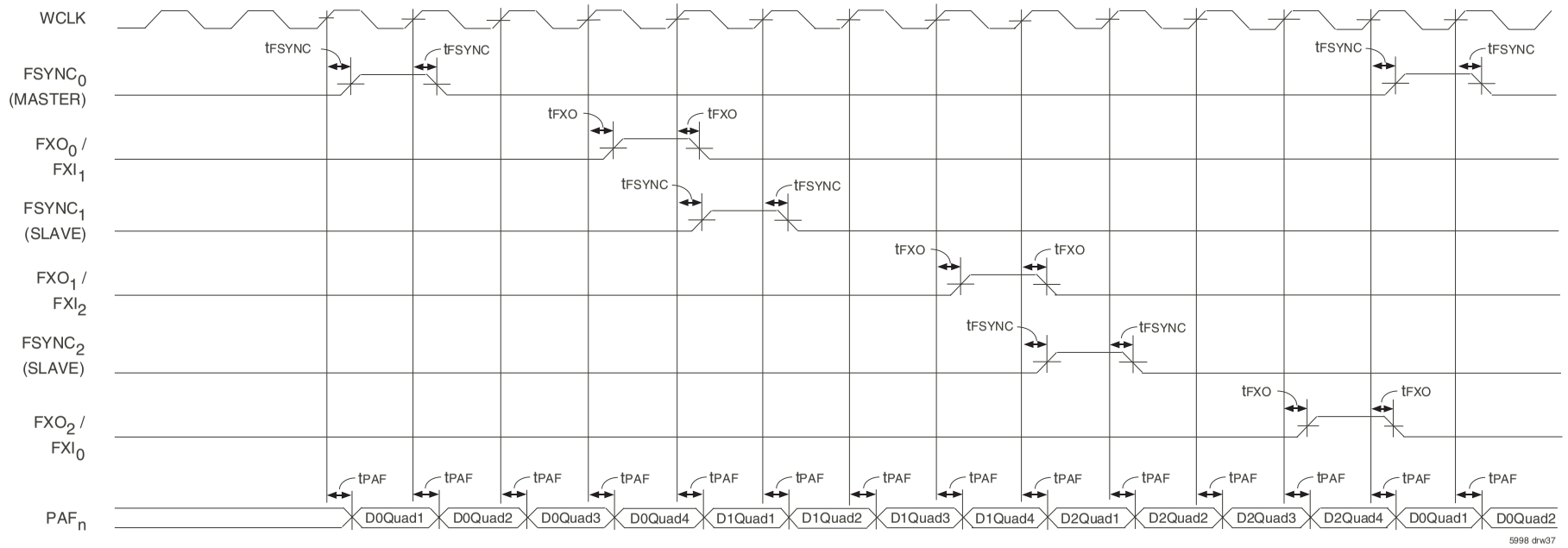
G Word, Wd-m+2 is read out due to FWFT operation.

GG $\overline{PAF}$[7] and the discrete $\overline{PAF}$ flag go LOW to show the write on cycle *DD* causes Q31 of D0 to again go almost full.
Word, Wy+2 is written into D0 Q31.

H No read operation.

I Word, W0 is read from Q0 of D6, selected on cycle *F*, due to FWFT.

Figure 31. $\overline{PAF}$n - Direct Mode, Flag Operation



NOTE:

1. This diagram is based on 3 devices connected in expansion mode.

Figure 32. $\overline{PAF}$n Bus - Polled Mode

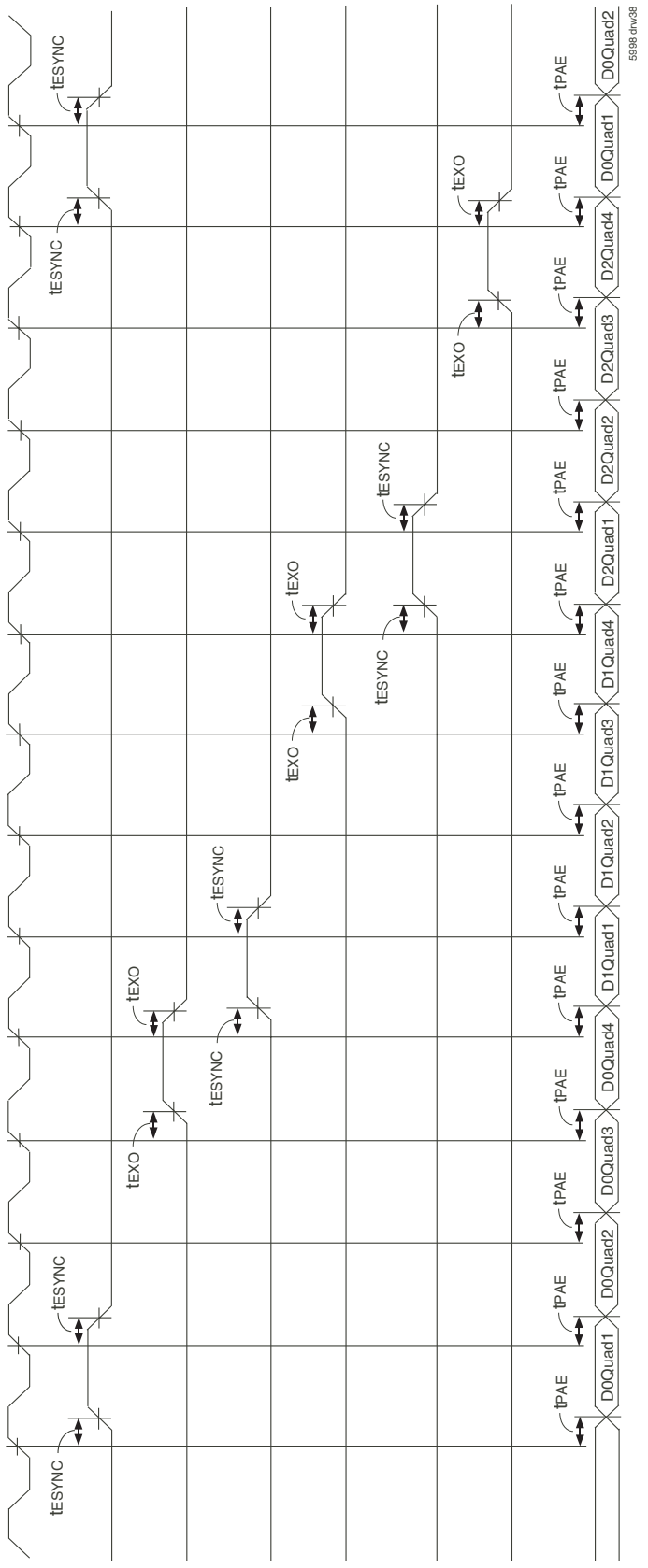
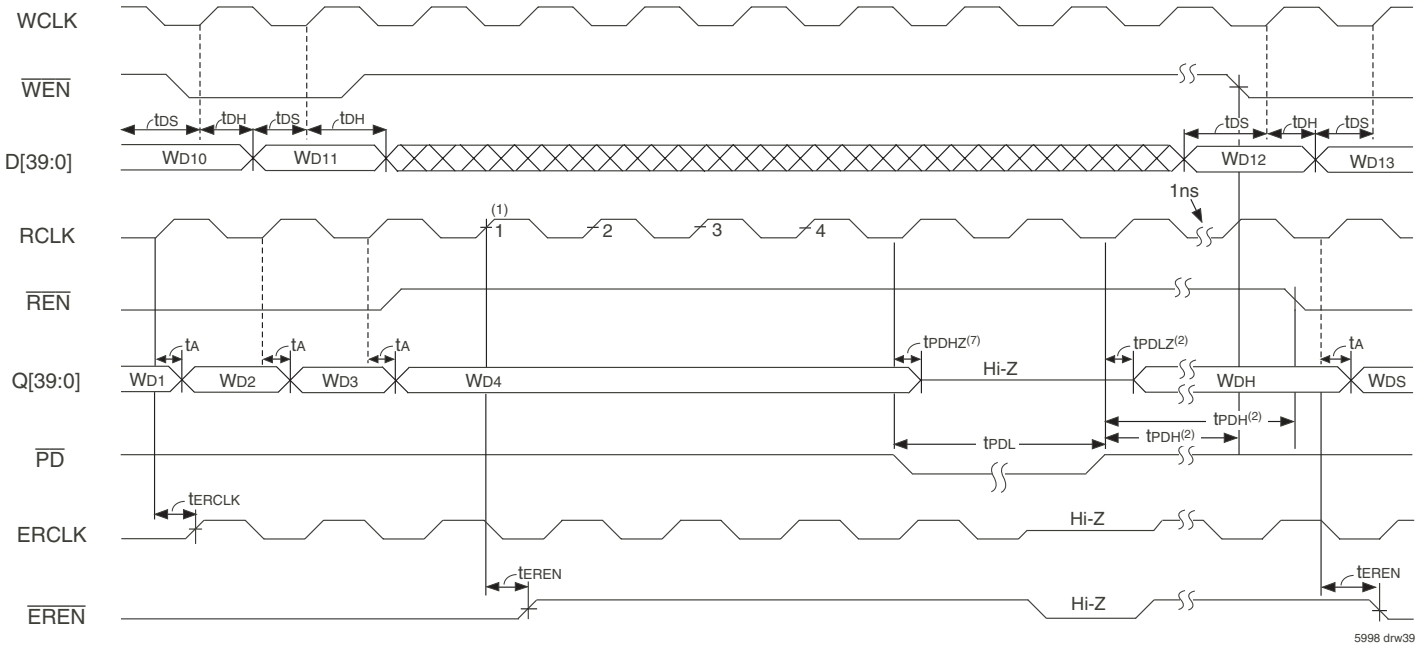


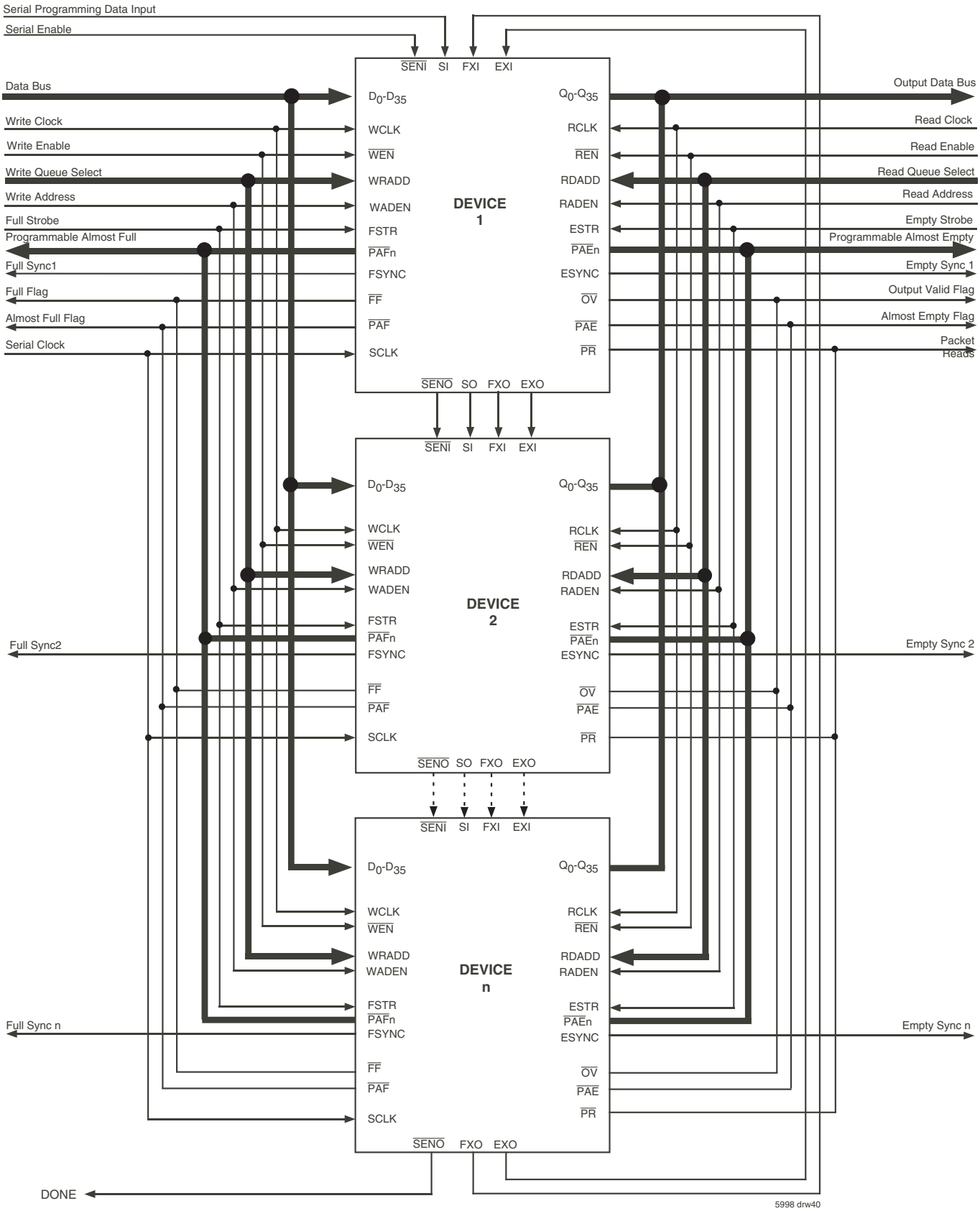
Figure 33. $\overline{PAE}n/\overline{PR}n$ Bus - Polled Mode

NOTE:
1. This diagram is based on 3 devices connected in expansion mode.



NOTES:
1. All read and write operations must have ceased a minimum of 4 WCLK and 4 RCLK cycles before power down is asserted.
2. When the $\overline{PD}$ input becomes deasserted, there will be a 1μs waiting period before read and write operations can resume. All input and output signals will also resume after this time period.
3. Set-up and configuration static inputs are not affected during power down.
4. Serial programming and JTAG programming port are inactive during power down.
5. $\overline{RCS}$ = 0, $\overline{WCS}$ = 0 and $\overline{OE}$ = 0. These signals can toggle during and after power down.
6. All flags remain active and maintain their current states.
7. During power down, all outputs will be in high-impedance.

Figure 34. Power Down Operation



NOTES:

1. If devices are configured for Direct operation of the $\overline{PAF}$n/$\overline{PAE}$n flag busses the FXI/EXI of the MASTER device should be tied LOW. All other devices tied HIGH. The FXO/EXO outputs are DNC (Do Not Connect).
2. Q outputs must not be mixed between devices, i.e. Q0 of device 1 must connect to Q0 of device 2, etc.

Figure 35. Multi-Queue Expansion Diagram

JTAG INTERFACE

Five additional pins (TDI, TDO, TMS, TCK and $\overline{TRST}$) are provided to support the JTAG boundary scan interface. The IDT72T51546/72T51556 incorporates the necessary tap controller and modified pad cells to implement the JTAG facility.

Note that IDT provides appropriate Boundary Scan Description Language program files for these devices.

The Standard JTAG interface consists of four basic elements:

- *Test Access Port (TAP)*
- *TAP controller*
- *Instruction Register (IR)*
- *Data Register Port (DR)*

The following sections provide a brief description of each element. For a complete description refer to the IEEE Standard Test Access Port Specification (IEEE Std. 1149.1-1990).

The Figure below shows the standard Boundary-Scan Architecture

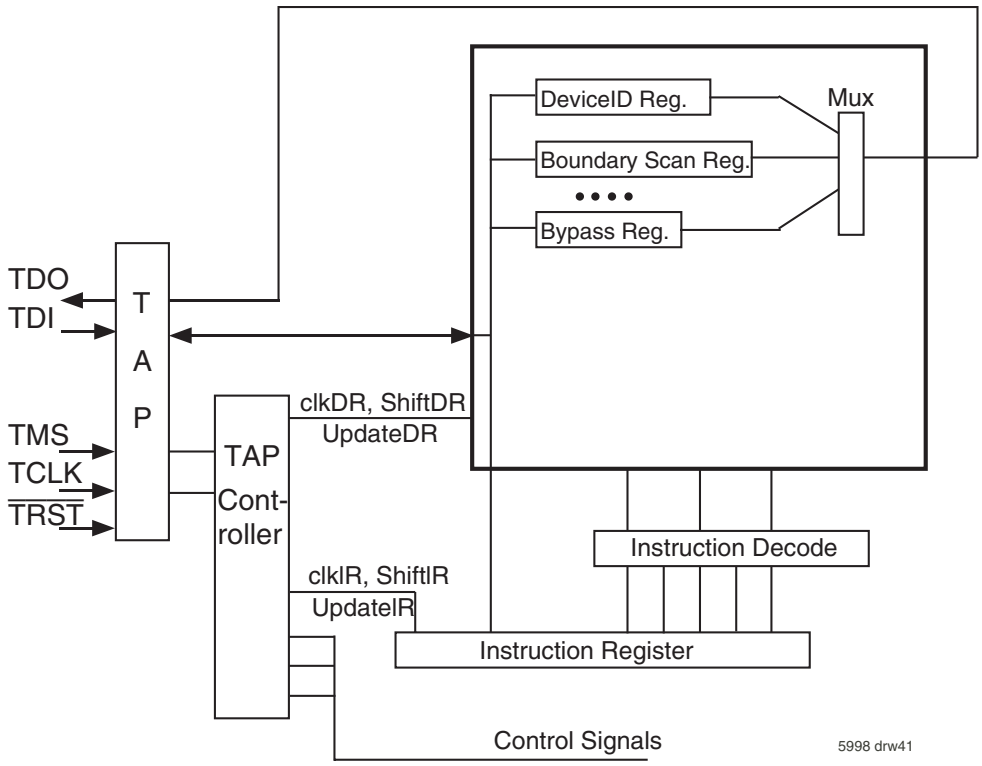


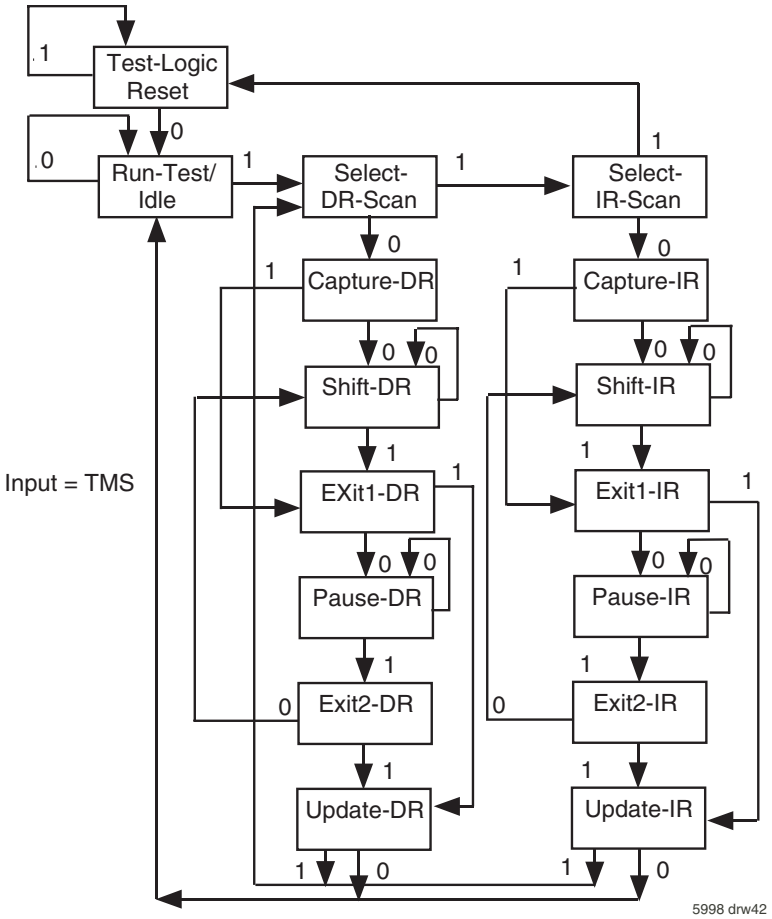
Figure 36. Boundary Scan Architecture

TEST ACCESS PORT (TAP)

The Tap interface is a general-purpose port that provides access to the internal of the processor. It consists of four input ports (TCLK, TMS, TDI, $\overline{TRST}$) and one output port (TDO).

THE TAP CONTROLLER

The Tap controller is a synchronous finite state machine that responds to TMS and TCLK signals to generate clock and control signals to the Instruction and Data Registers for capture and update of data.



NOTES:
1. Five consecutive TCK cycles with TMS = 1 will reset the TAP.
2. TAP controller does not automatically reset upon power-up. The user must provide a reset to the TAP controller (either by $\overline{TRST}$ or TMS).
3. TAP controller must be reset before normal Queue operations can begin.

Figure 37. TAP Controller State Diagram

Refer to the IEEE Standard Test Access Port Specification (IEEE Std. 1149.1) for the full state diagram.

All state transitions within the TAP controller occur at the rising edge of the TCLK pulse. The TMS signal level (0 or 1) determines the state progression that occurs on each TCLK rising edge. The TAP controller takes precedence over the Queue and must be reset after power up of the device. See $\overline{TRST}$ description for more details on TAP controller reset.

Test-Logic-Reset All test logic is disabled in this controller state enabling the normal operation of the IC. The TAP controller state machine is designed in such a way that, no matter what the initial state of the controller is, the Test-Logic-Reset state can be entered by holding TMS at high and pulsing TCK five times. This is the reason why the Test Reset ($\overline{TRST}$) pin is optional.

Run-Test-Idle In this controller state, the test logic in the IC is active only if certain instructions are present. For example, if an instruction activates the self test, then it will be executed when the controller enters this state. The test logic in the IC is idles otherwise.

Select-DR-Scan This is a controller state where the decision to enter the Data Path or the Select-IR-Scan state is made.

Select-IR-Scan This is a controller state where the decision to enter the Instruction Path is made. The Controller can return to the Test-Logic-Reset state other wise.

Capture-IR In this controller state, the shift register bank in the Instruction Register parallel loads a pattern of fixed values on the rising edge of TCK. The last two significant bits are always required to be "01".

Shift-IR In this controller state, the instruction register gets connected between TDI and TDO, and the captured pattern gets shifted on each rising edge of TCK. The instruction available on the TDI pin is also shifted in to the instruction register.

Exit1-IR This is a controller state where a decision to enter either the Pause-IR state or Update-IR state is made.

Pause-IR This state is provided in order to allow the shifting of instruction register to be temporarily halted.

Exit2-DR This is a controller state where a decision to enter either the Shift-IR state or Update-IR state is made.

Update-IR In this controller state, the instruction in the instruction register is latched in to the latch bank of the Instruction Register on every falling edge of TCK. This instruction also becomes the current instruction once it is latched.

Capture-DR In this controller state, the data is parallel loaded in to the data registers selected by the current instruction on the rising edge of TCK.

Shift-DR, Exit1-DR, Pause-DR, Exit2-DR and Update-DR These controller states are similar to the Shift-IR, Exit1-IR, Pause-IR, Exit2-IR and Update-IR states in the Instruction path.

THE INSTRUCTION REGISTER

The Instruction register allows an instruction to be shifted in serially into the processor at the rising edge of TCLK.

The Instruction is used to select the test to be performed, or the test data register to be accessed, or both. The instruction shifted into the register is latched at the completion of the shifting process when the TAP controller is at Update-IR state.

The instruction register must contain 4 bit instruction register-based cells which can hold instruction data. These mandatory cells are located nearest the serial outputs they are the least significant bits.

TEST DATA REGISTER

The Test Data register contains three test data registers: the Bypass, the Boundary Scan register and Device ID register.

These registers are connected in parallel between a common serial input and a common serial data output.

The following sections provide a brief description of each element. For a complete description, refer to the IEEE Standard Test Access Port Specification (IEEE Std. 1149.1-1990).

TEST BYPASS REGISTER

The register is used to allow test data to flow through the device from TDI to TDO. It contains a single stage shift register for a minimum length in serial path. When the bypass register is selected by an instruction, the shift register stage is set to a logic zero on the rising edge of TCLK when the TAP controller is in the Capture-DR state.

The operation of the bypass register should not have any effect on the operation of the device in response to the BYPASS instruction.

THE BOUNDARY-SCAN REGISTER

The Boundary Scan Register allows serial data TDI be loaded in to or read out of the processor input/output ports. The Boundary Scan Register is a part of the IEEE 1149.1-1990 Standard JTAG Implementation.

THE DEVICE IDENTIFICATION REGISTER

The Device Identification Register is a Read Only 32-bit register used to specify the manufacturer, part number and version of the processor to be determined through the TAP in response to the IDCODE instruction.

IDT JEDEC ID number is 0xB3. This translates to 0x33 when the parity is dropped in the 11-bit Manufacturer ID field.

For the IDT72T51546/72T51556, the Part Number field contains the following values:

Device	Part# Field (HEX)
IDT72T51546	0x48C
IDT72T51556	0x48D

31(MSB) 28	27 12	11 1	0(LSB)
Version (4 bits) 0X0	Part Number (16-bit)	Manufacturer ID (11-bit) 0X33	1

JTAG DEVICE IDENTIFICATION REGISTER

JTAG INSTRUCTION REGISTER

The Instruction register allows instruction to be serially input into the device when the TAP controller is in the Shift-IR state. The instruction is decoded to perform the following:

- Select test data registers that may operate while the instruction is current. The other test data registers should not interfere with chip operation and the selected data register.
- Define the serial test data register path that is used to shift data between TDI and TDO during data register scanning.

The Instruction Register is a 4 bit field (i.e. IR3, IR2, IR1, IR0) to decode 16 different possible instructions. Instructions are decoded as follows.

Hex Value	Instruction	Function
00	EXTEST	Select Boundary Scan Register
01	SAMPLE/PRELOAD	Select Boundary Scan Register
02	IDCODE	Select Chip Identification data register
04	HIGH-IMPEDANCE	JTAG
0F	BYPASS	Select Bypass Register

JTAG INSTRUCTION REGISTER DECODING

The following sections provide a brief description of each instruction. For a complete description refer to the IEEE Standard Test Access Port Specification (IEEE Std. 1149.1-1990).

EXTEST

The required EXTEST instruction places the IC into an external boundary-test mode and selects the boundary-scan register to be connected between TDI and TDO. During this instruction, the boundary-scan register is accessed to drive test data off-chip via the boundary outputs and receive test data off-chip via the boundary inputs. As such, the EXTEST instruction is the workhorse of IEEE. Std 1149.1, providing for probe-less testing of solder-joint opens/shorts and of logic cluster function.

IDCODE

The optional IDCODE instruction allows the IC to remain in its functional mode and selects the optional device identification register to be connected between TDI and TDO. The device identification register is a 32-bit shift register containing information regarding the IC manufacturer, device type, and version code. Accessing the device identification register does not interfere with the operation of the IC. Also, access to the device identification register should be immediately available, via a TAP data-scan operation, after power-up of the IC or after the TAP has been reset using the optional $\overline{\text{TRST}}$ pin or by otherwise moving to the Test-Logic-Reset state.

SAMPLE/PRELOAD

The required SAMPLE/PRELOAD instruction allows the IC to remain in a normal functional mode and selects the boundary-scan register to be connected between TDI and TDO. During this instruction, the boundary-scan register can be accessed via a date scan operation, to take a sample of the functional data entering and leaving the IC. This instruction is also used to preload test data into the boundary-scan register before loading an EXTEST instruction.

HIGH-IMPEDANCE

The optional High-Impedance instruction sets all outputs (including two-state as well as three-state types) of an IC to a disabled (high-impedance) state and selects the one-bit bypass register to be connected between TDI and TDO. During this instruction, data can be shifted through the bypass register from TDI to TDO without affecting the condition of the IC outputs.

BYPASS

The required BYPASS instruction allows the IC to remain in a normal functional mode and selects the one-bit bypass register to be connected between TDI and TDO. The BYPASS instruction allows serial data to be transferred through the IC from TDI to TDO without affecting the operation of the IC.

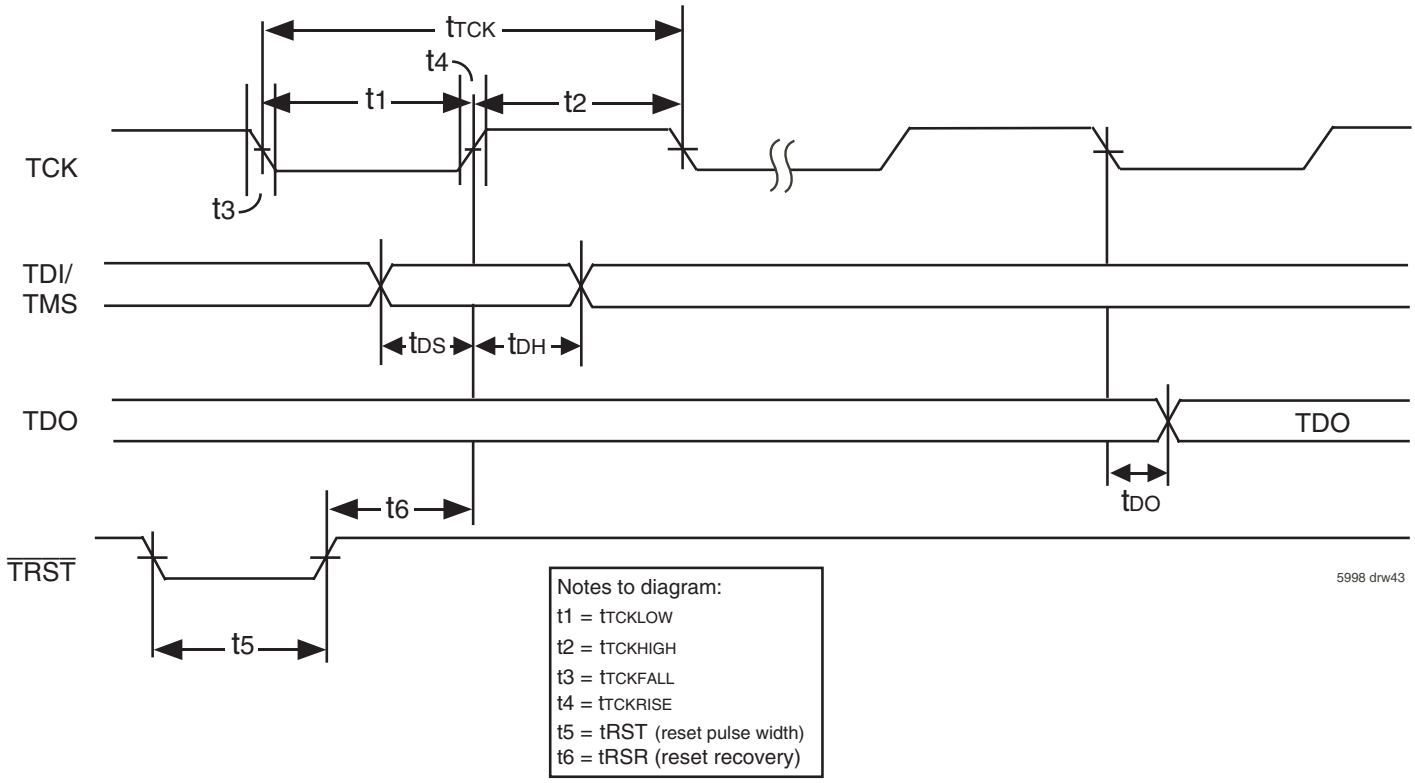


Figure 38. Standard JTAG Timing

SYSTEM INTERFACE PARAMETERS

Parameter	Symbol	Test Conditions	IDT72T51546 IDT72T51556		
			Min.	Max.	Units
Data Output	t_{DO}[1]		-	20	ns
Data Output Hold	t_{DOH}[1]		0	-	ns
Data Input	t_{DS}	trise=3ns	10	-	ns
	t_{DH}	tfall=3ns	10	-	

NOTE:
1. 50pf loading on external output signals.

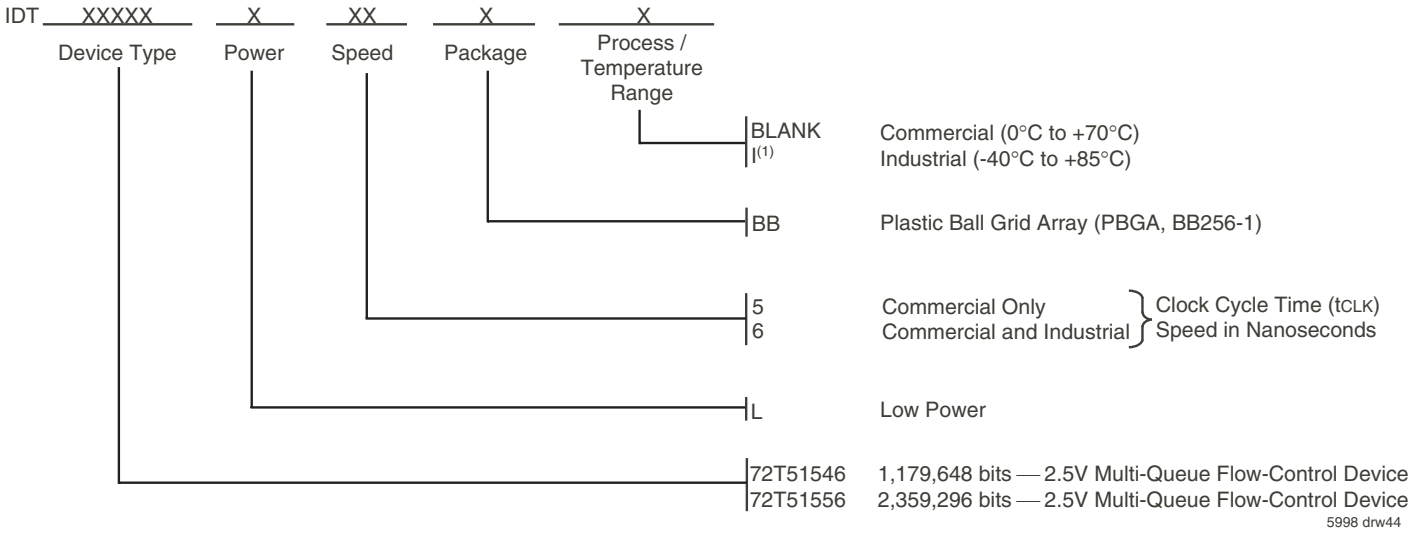
JTAG AC ELECTRICAL CHARACTERISTICS

(Vcc = 2.5V ± 5%; Tcase = 0°C to +85°C)

Parameter	Symbol	Test Conditions			
			Min.	Max.	Units
JTAG Clock Input Period	tTCK	-	100	-	ns
JTAG Clock HIGH	tTCKHIGH	-	40	-	ns
JTAG Clock Low	tTCKLOW	-	40	-	ns
JTAG Clock Rise Time	tTCKRISE	-	-	5[1]	ns
JTAG Clock Fall Time	tTCKFALL	-	-	5[1]	ns
JTAG Reset	tRST	-	50	-	ns
JTAG Reset Recovery	tRSR	-	50	-	ns

NOTE:
1. Guaranteed by design.

ORDERING INFORMATION



NOTE:
1. Industrial temperature range product for the 6ns is available as a standard device. All other speed grades available by special order.

DATASHEET DOCUMENT HISTORY

06/06/2003 pgs. 1 through 64.
11/06/2003 pgs. 1, 4, 17 and 18.

CORPORATE HEADQUARTERS
2975 Stender Way
Santa Clara, CA 95054

for SALES:
800-345-7015 or 408-727-6116
fax: 408-492-8674
www.idt.com

for Tech Support:
408-330-1533
email: Flow-Controlhelp@idt.com